NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

IMPROVING THE PERFORMANCE OF MEMS GYROS VIA REDUNDANT MEASUREMENTS: THEORY AND EXPERIMENTS

by

Matthew J. Leszczynski

December 2014

Thesis Advisor:	Mark Karpenko
Second Reader:	I. Michael Ross

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 2014	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE IMPROVING THE PERFORMANCE OF MEMS GYROS VIA REDUNDANT MEASUREMENTS: THEORY AND EXPERIMENTS		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Matthew J. Leszczynski			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE A

13. ABSTRACT (maximum 200 words)

A redundant sensor array is a device composed of more than one sensor of a given type, for example an array of rate gyroscopes. Previous work has shown that redundancy can enhance the failure tolerance of the sensor suite and can also be leveraged to improve the accuracy of the measurements. This thesis further explores the possibility of improving the performance of low-cost micro-electro-mechanical (MEMS) gyroscopes using the redundancy concept. To this end, an experimental sensor array test bed is constructed that allows data from up to 12 three degrees of freedom MEMS gyroscopes to be acquired and combined under various geometric configurations. By organizing the MEMS gyroscopes in ideal geometric configurations, it is shown how the effects of sensor noise can be reduced in order to increase the precision of the measurements. Experiments using the constructed sensor array test bed show that the 1-σ sensor noise can be reduced by nearly 65 percent through the proper combination of multiple measurements using ten 3 degrees of freedom gyros. It is further observed that realizing this improvement relies, in part, upon accurate estimation of the sensor bias. This aspect motivates future work on bias estimation schemes for arrays of sensors.

14. SUBJECT TERMS optimal redundant sensor, micro-electro-mechanical systems (MEMS) gyroscope arrays, improve performance inertial measurement unit (IMU), Sparkfun razor IMU, gyroscope, magnetometer, accelerometer, redundant IMU, angular rate accuracy (ARW)			**15. NUMBER OF PAGES** 101
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

IMPROVING THE PERFORMANCE OF MEMS GYROS VIA REDUNDANT MEASUREMENTS: THEORY AND EXPERIMENTS

Matthew J. Leszczynski
Lieutenant, United States Navy
B.S.EE, San Diego State University, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2014

Author: Matthew J. Leszczynski

Approved by: Mark Karpenko
                 Thesis Advisor

                 I. Michael Ross
                 Second Reader

                 Garth V. Hobson
                 Chair, Department of Mechanical and Aerospace Engineering

iii

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A redundant sensor array is a device composed of more than one sensor of a given type, for example an array of rate gyroscopes. Previous work has shown that redundancy can enhance the failure tolerance of the sensor suite and can also be leveraged to improve the accuracy of the measurements. This thesis further explores the possibility of improving the performance of low-cost micro-electro-mechanical (MEMS) gyroscopes using the redundancy concept. To this end, an experimental sensor array test bed is constructed that allows data from up to 12 three degrees of freedom MEMS gyroscopes to be acquired and combined under various geometric configurations. By organizing the MEMS gyroscopes in ideal geometric configurations, it is shown how the effects of sensor noise can be reduced in order to increase the precision of the measurements. Experiments using the constructed sensor array test bed show that the 1-σ sensor noise can be reduced by nearly 65 percent through the proper combination of multiple measurements using ten 3 degrees of freedom gyros. It is further observed that realizing this improvement relies, in part, upon accurate estimation of the sensor bias. This aspect motivates future work on bias estimation schemes for arrays of sensors.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

xii

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADC	analog to digital converter
ARW	angle random walk
CAD	computer assisted design
cRIO	compact reconfigurable input output
DOF	degrees of freedom
EPS	electrical power system
FPGA	field programmable gate array
IMU	inertial measurement unit
LabVIEW	laboratory virtual instrument engineering workbench
MEMS	micro-electro-mechanical systems
NI	National Instruments
TCP	torsion control plant
UART	universal asynchronous receiver transmitter
USB	universal serial bus
VI	virtual instrument

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

THIS PAGE INTENTIONALLY LEFT BLANK

I.   INTRODUCTION

A.   MOTIVATION

An inertial measurement unit (IMU) measures linear and angular motion in three dimensions without external reference [1]. In order to accomplish this, the IMU uses a combination of accelerometers, gyroscopes, and magnetometers. IMU outputs can be angles, accelerations, and angular rates. These are all measured in relation to the inertial sensor frame.

IMUs are used to estimate position, velocity, and attitude in a wide range of applications. However, the high cost of military grade IMUs has led to increased focus on finding more affordable options. This has motivated development of micro-electro-mechanical systems (MEMS) IMUs. Compared to traditional IMUs, MEMS IMUs offer similar functionality but at a reduced cost, size, and power consumption. Currently, low-cost MEMS IMUs are successfully used in applications where their comparatively low precision can be largely ignored [1]. For example, the sensors used in MEMS IMUs are commonly found in applications such as screen rotation sensors for smart phones, portable gaming platforms, and robotic guidance.

Because of the cost-saving potential associated with MEMS, consumers who would normally pay a premium for traditional high-end sensors are now searching instead for methods to improve the performance of MEMS devices. Previous research has focused on improving the precision and accuracy of the MEMS sensors themselves, while very little research has been done to investigate improving the performance by other means [2]. For example, MEMS IMUs can be arranged in various geometrical configurations to increase precision and redundancy. This thesis presents an investigation into improving the precision of a redundant MEMS gyro array by finding the ideal orientation of its individual sensors.

B.   MEMS INERTIAL MEASUREMENT UNITS

MEMS technology can be used to produce complex structures, devices and systems. MEMS refer to devices that are a combination of electrical and mechanical

components that are typically fabricated using integrated circuit batch processing techniques [2]. Most signals of interest that occur when measuring aspects of the physical world are analog signals. Transduction mechanisms are used to convert real-world signals from one form of energy to another. For example, a thermocouple converts a change in temperature to an analog voltage, while an inertial sensor converts measured acceleration or angular rate to an analog voltage signal [3]. Nearly all IMUs incorporate the use of a gyroscope [5]; therefore, this thesis will ignore the accelerometer and magnetometer sensors of the IMU and strictly focus on the gyroscope.

1. MEMS Gyroscope

A gyroscope is a device used primarily for measurement of angular velocity. An example of how a gyro could be used is depicted in Figure 1. A 3-axis gyroscope can simultaneously sense changes in the yaw, pitch, and roll axes. If the turn table was rotating at 10 rpm about the yaw axis, the gyro would measure a constant rotation of 360° times 10 rpm divided by 60 seconds (i.e., 60°/s). The gyro would then output a voltage proportional to this angular rate. The voltage produced would be determined by the sensitivity of the gyro, which is typically measured in millivolts per degree per second (mV/°/s) [4].
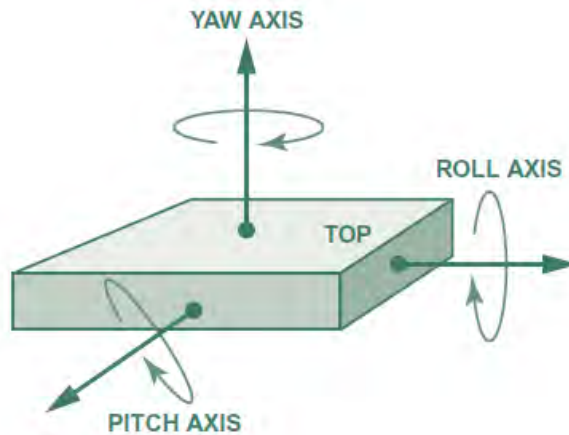
Figure 1   Three-axis gyroscope depicting all axes of sensitivity, from [4].

Gyroscopes have evolved from mechanical-inertial spinning devices that consisted of rotors, axles, and gimbals to combinations of electronic and optical devices. Each method of design takes advantage of a physical property of the system that allows it to detect rotational velocity about a measurement axis. There are three basic types of gyroscopes:

- Rotary (classical) gyroscopes [5]
- MEMS (vibrating structure) gyroscope [4, 6, 7]
- Optical gyroscopes [8]

To understand the operation of a MEMS gyroscope it is necessary to understand the concept of Coriolis acceleration. MEMS gyros measure angular rate by means of Coriolis acceleration. The Coriolis acceleration can be explained with the help of Figure 2.
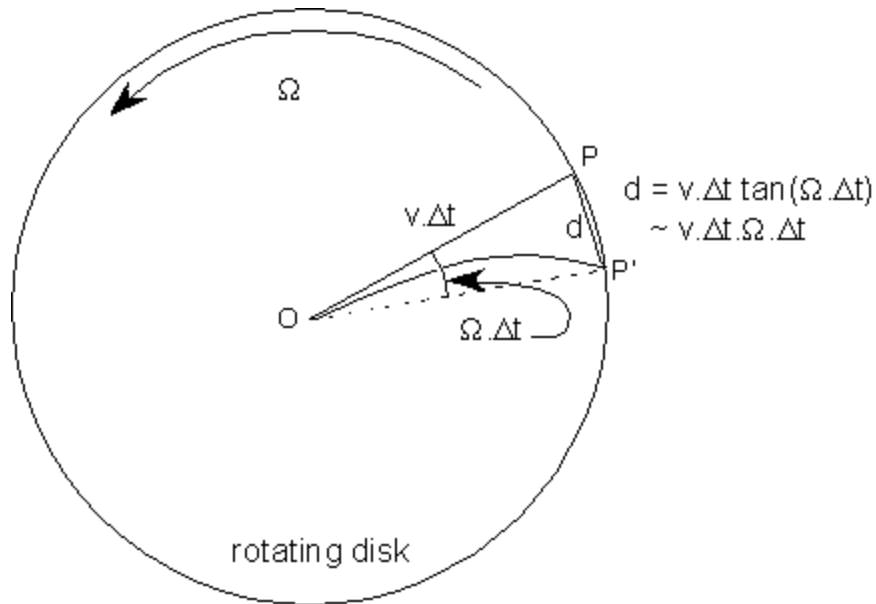
Figure 2   Coriolis acceleration example, from [9].

Consider an object travelling from the center (point O) towards P on the edge of the rotating disk depicted in Figure 2. It takes time Δt when travelling at speed v, so the distance OP equals $v\Delta t$. However, after a time Δt, P will have moved from its original position (P') to a new position (P). The angle POP' is $\Omega\Delta t$. Where Ω is the speed of

3

rotation of the disk (in radians per second). For very short displacements, using small angle approximations, the distance, $d = $ PP' is:

$$d \approx v\Delta t^2 \Omega \qquad (1.1)$$

The motion of point P' to point P through length d results from the Coriolis acceleration, a_c. The distance, d, between P' and P can be expressed as,

$$d = a_c \frac{\Delta t^2}{2} \qquad (1.2)$$

Combining (1.1) and (1.2) gives the Coriolis acceleration as:

$$a_c = 2v\Omega \qquad (1.3)$$

Multiply the acceleration by a mass to get the Coriolis force:

$$F_c = 2v\Omega m \qquad (1.4)$$

MEMS gyros take advantage of Coriolis acceleration by using a vibrating mass that moves in and out on a rotating platform. The mass is micro-machined from poly-silicon and is tethered to a poly-silicon frame so that it travels only along one direction. Just like the example above, Figure 3(a) shows that when the resonating mass, m, moves toward the outer edge of the rotation, as indicated by the red arrow, it is accelerated to the right. This exerts a reaction force on the frame to the left, given by (1.4), as shown by the blue arrow. When the mass moves toward the center of rotation in Figure 3(b), it exerts a force to the right, as indicated by the blue arrow.
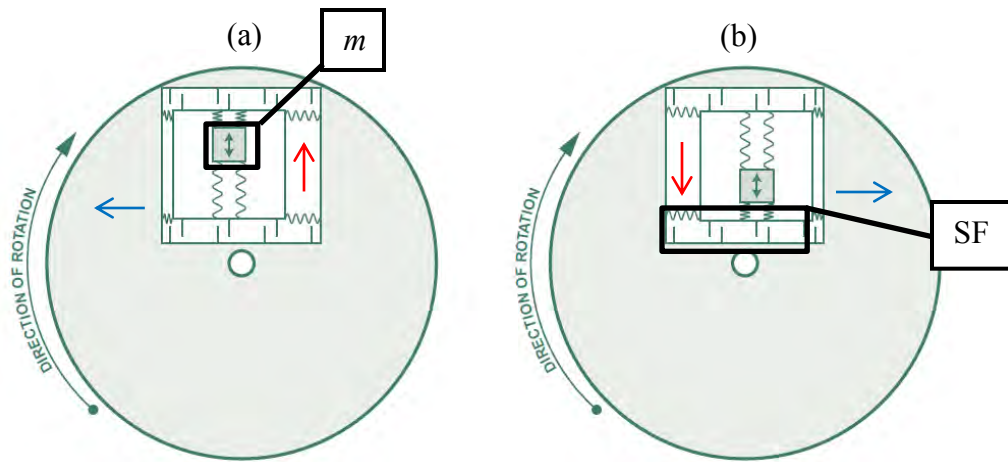
Figure 3   Coriolis effect on a MEMS vibrating mass. The frame and vibrating mass (*m*) are displaced laterally by the Coriolis effect. The displacement is determined from the change in capacitance between the sense fingers, after [4].

To measure the Coriolis acceleration, the frame containing the vibrating mass is tethered to the substrate by springs at 90º relative to the resonating motion. This is shown in greater detail in Figure 4.
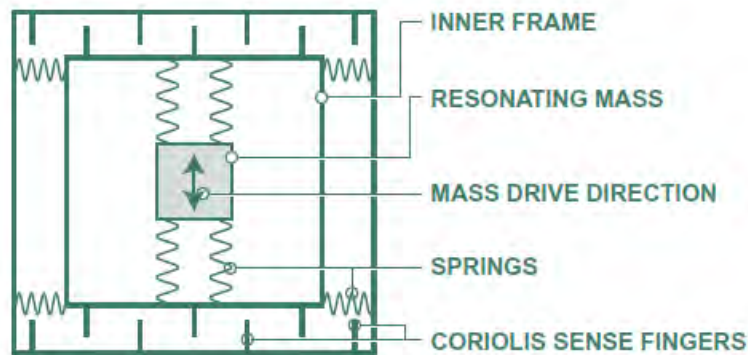
Figure 4   Schematic of a MEMS gyro mechanical structure, from [4].

Figure 4 also shows the sensing fingers which use capacitance to sense the displacement of the frame in response to the force exerted by the vibrating mass [4]. If the springs have stiffness K, then the displacement resulting from the reaction force will be

$$\delta = 2\Omega v \frac{m}{K} \qquad\qquad (1.5)$$

The voltage across the capacitive sensing fingers is a function of the distance between the sensing fingers and can be expressed as

$$C = \frac{\varepsilon A}{\delta} \qquad\qquad (1.6)$$

The voltage across the capacitor is expressed as

$$V = \frac{Q}{C} \qquad\qquad (1.7)$$

where:

- C=Capacitance
- ε=Permittivity of dielectric
- A=Area of conductive plates
- δ=Distance between plates
- Q=Charge of the capacitor

Therefore, the voltage is proportional to the angular rate:

$$V = 2\frac{QVm}{K\varepsilon A}\Omega \qquad\qquad (1.8)$$

The voltage is normally output in a quantized fashion by the analog to digital converter (ADC) to provide a digital output that can be identified by other devices.

2.   Limitations of MEMS Gyros

Since the initial introduction of MEMS gyroscopes over two decades ago [10], their resolution has improved by an approximate factor of 10 every two years [11]. Current venders of state-of-the-art MEMS gyroscopes report a bias instability of less than 10º/hr for a majority of their high-end products [7]. The bias of a MEMS gyro will wander or walk over time due to noise in the electronics and other effects. The bias fluctuations due to noise are typically modeled as angle random walk. A bias stability measurement tells you how stable the bias of a gyro is over a certain specified period of time. The value given in gyro datasheets is typically given in units of deg/hr or deg/s for MEMS gyros. Despite the continuing improvement in the angular rate resolution

achieved by MEMS gyros, the noise inherent to MEMS sensor design causes reductions in precision due to bias drift effects. This issue limits the use of MEMS technology in field applications that require high precision and accuracy.

The precision of MEMS gyros depends primarily on the high frequency noise caused by thermo-mechanical events, which are related to their construction. This type of noise generally shows up as white noise. The white noise adds a random rate signal to the output signal with a finite variance and zero long term mean [12].

Reducing noise effects is important because when a MEMS gyroscope is used to track the orientation of an object, its output signal must be integrated over time. Since the output signal is corrupted by noise, a buildup of orientation angle drift occurs. This drift is referred to in industry as angle random walk (ARW). In order to have a high precision MEMS-based inertial measurement, a stable, low-noise, MEMS gyroscope must be developed.

C.   COMBINING MULTIPLE INERTIAL MEASUREMENT UNITS

The advantages of using multiple sensors to improve the precision of an array have been recognized and employed in many facets of industry. Published papers document a variety of different methods to use sensor redundancy, sensor configuration, or signal processing techniques to achieve increased precision over a single sensor. This section presents several of these papers and explains how they are related to this thesis.

Researchers at Northwestern Polytechnic University published an open access article [12] in which the authors describe the idea of creating a "virtual gyroscope". This "virtual gyroscope" is based on the principle that the output of several identical low-precision gyroscopes in an array can be combined together to more precisely measure the state of an object [12]. This paper used analytical models to provide synthetic inputs to the virtual gyro. An optimal filter was used to improve the precision beyond the performance limitations of the individual gyroscopes. Much like the idea behind this "virtual gyroscope," the possibility exists to further improve the overall precision by adding more sensors in an optimal geometric configuration. This thesis moves past the

analytical modeling, presented in [12], into actual experiments that attempt to validate the theory using real sensors.

The first step towards this validation is to explore how the precision of a simple, two-dimensional sensor array can be improved by altering its geometry. A successful proof of concept leads to the experimental work performed in later chapters of this thesis. The end goal is to build on the concepts developed by Jafari and Roshanian [13]. In their article, the authors analyze the precision of redundant IMUs under various configurations. Rather than improving the precision, their focus is on the reliability that is achieved through redundancy. The derivation of a least-squares based transformation matrix is applied to the output of any combination of sensors to obtain orthogonal vector components based on their configuration. This result which allows the analytic variance of the sensor array to be determined is the basis for this thesis. The specifics of which will be detailed in later chapters.

A related work, [14], provides efficient and low-cost solutions to increasing the precision of an IMU by using different configurations of redundant MEMS sensors. Through modeling in MATLAB, it is possible to demonstrate improved array precision for several array configurations. The results shown in [14] study several array configurations, as well as methods by which to compare them against each other. Similar methods are employed in this thesis for comparing the advantages of multiple sensors.

The process of analyzing and modeling raw IMU data, as well as how to use the results of that analysis to design an inertial navigation system (INS), is described in [15]. Part of the focus is on understanding and characterizing the type of noise that is commonly associated with MEMS IMUs. The same IMU made by Sparkfun that is examined in [15] is used in this thesis. Therefore, the methods used to characterize the IMU noise signals in [15] provide a benchmark for the results of this thesis.

D. RESEARCH CONTRIBUTION

The research completed in this thesis augments the results of the papers discussed in the last section. Primarily, this thesis moves beyond the analytical and numerical results obtained through modeling, towards experimental validation of the concepts

8

presented in previous papers. This thesis also develops an embedded hardware system that can produce real data and can therefore be used to validate the analytic results of previous papers. Parts of this thesis also explore how to determine an ideal configuration for a subset of a given number of sensors arranged in a given geometry, thus, contributing further to the literature.

E. THESIS OUTLINE

This thesis is laid out in the same order that the research was conducted. The following outlines the research that was accomplished, and the structure of this thesis:

- Validate previous optimal redundant sensor research using analytical MATLAB models.

- Create a test bed that allows for the experimental validation of the analytical results obtained.

- Use two methods to collect test data, first using serial data logging software, and then using a data acquisition computer.

- Process data to determine the precision of the new redundant gyro array.

The chapters of the thesis are briefly described below.

- Chapter II focuses on a MATLAB study aimed at recreating aspects of the work done in [13] to validate the ability to affect the precision of a sensor array by altering the configuration and number of sensors in the array.

- Chapter III presents the construction of the sensor array test bed, and introduces the hardware and software used to collect data from the IMUs.

- Chapter IV presents initial testing and validation of the Sparkfun razor IMU and the systems used to acquire and manage the raw data.

- Chapter V presents analysis of initial experiments and full scale testing using the designed sensor array.

- Chapter VI summarizes the results and contribution then outlines potential ideas for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II.   IMPROVING PRECISION THROUGH REDUNDANCY

A.   INTRODUCTION

The purpose of this chapter is to illustrate how multiple sensors can be used together to improve the precision of a measurement. To do this an analytical model is created to simulate a two sensor gyro array. An example of a two sensor gyro array is shown in Figure 5. The red and blue arrows represent the measurement axes of two independent MEMS rate gyroscopes. The two elevation angles EL^1 and EL^2 can be varied in order to create an ideal configuration. This chapter illustrates how these angles can be optimally selected and how their values influence the precision of the resulting sensor array.
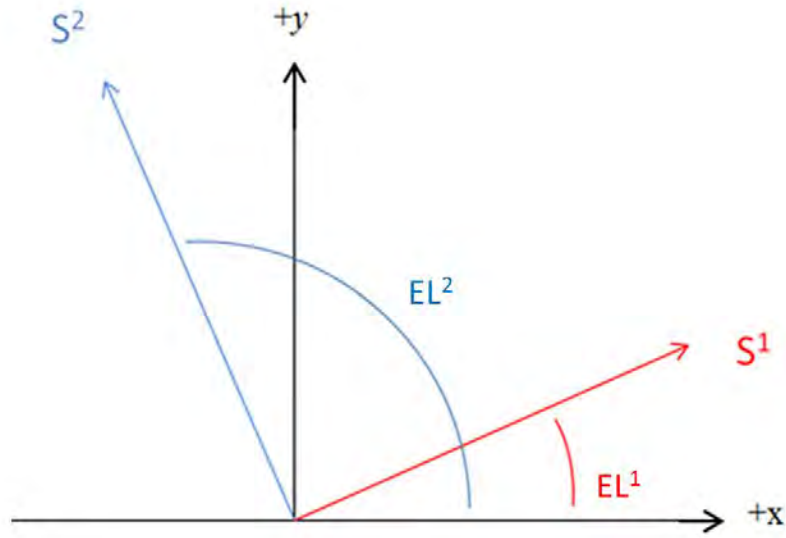
Figure 5   Two sensors with their measurement axes aligned along S^1 and S^2

B.   ANALYSIS

Showing that there is a possibility of improving the precision of a group of sensors starts with defining a generic sensor orientation. The sensor orientation can be used to define the configuration of the entire multi-sensor array. One approach is to define the orientation of each senor axis using azimuth and elevation angles with respect

to an orthogonal reference frame. This is accomplished by using a coordinate transformation to rotate the individual sensor coordinate systems to match that of the world coordinate system. Equation (2.1) is a unit vector that describes the sensor axis S^i in a three dimensional orthogonal reference frame. The equation is developed by a series of projections along each of the three primary axes. Angle EL^i is the elevation angle of sensor i, and AZ^i is the azimuth [13].

$$S^i = \cos(EL^i)\cos(AZ^i)\hat{i} + \cos(EL^i)\sin(AZ^i)\hat{j} + \sin(EL^i)\hat{k} \tag{2.1}$$

Vector S^i is the unit vector along the sensing direction of the ith sensor expressed in the body-fixed coordinate system. For simplicity, this chapter analyzes the improvement in performance of only two sensors in a plane. Therefore, i=2 and equation (2.1) defining the sensing axis of each sensor simplifies to (2.2).

$$S^i = \cos(EL^i)\hat{i} + \sin(EL^i)\hat{j} \tag{2.2}$$

The geometrical arrangement of all the sensors can be combined in a matrix. This matrix is a collection of individual sensor unit vectors shown in (2.2). The collection is placed in matrix format to simplify later calculations. For the sensor arrangement of Figure 5, the sensor matrix, H, is defined as

$$H = \begin{bmatrix} S^1\hat{i} & S^1\hat{j} \\ S^2\hat{i} & S^2\hat{j} \end{bmatrix} = \begin{bmatrix} \cos(EL^1) & \sin(EL^1) \\ \cos(EL^2) & \sin(EL^2) \end{bmatrix} \tag{2.3}$$

It is necessary to add the random noise (V) to simulate the sensor noise associated with a real world system. This is done by using the "randn" function in MATLAB to generate random variables having a Gaussian distribution. Given this system of two sensors in two dimensions, the measurements achieved from each sensor can now be formulated as

$$m = H \cdot \omega + V \tag{2.4}$$

where:

- m^i is the resultant measurement of sensor i, $\omega = [\omega_x, \omega_y]$

12

- V is the simulated measurement noise. The noise is assumed to be white with a mean, $\mu=0$, and standard deviation, σ_i.

Equation (2.4) can also be expanded for any number of sensors in three dimensions as

$$\begin{bmatrix} m^1 \\ m^2 \\ \vdots \\ m^n \end{bmatrix} = \begin{pmatrix} S^1\hat{i} & S^1\hat{j} & S^1\hat{k} \\ S^2\hat{i} & S^2\hat{j} & S^2\hat{k} \\ \vdots & \vdots & \vdots \\ S^n\hat{i} & S^n\hat{j} & S^n\hat{k} \end{pmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \qquad (2.5)$$

where:

- ω_x, ω_y, and ω_z are the actual body rates body-fixed frame.

In order to estimate the true rate based on each sensor's measurement, matrix algebra is used to solve (2.4) for the estimated angular rate, $\hat{\omega}$. The result is shown in (2.6).

$$\hat{\omega} = H^{-1}(m+V) \qquad (2.6)$$

There are several methods inverting H, and the implications of each influence the results achieved. The Moore Penrose pseudo-inverse is commonly used cases where the matrix will not be square. The Moore Penrose pseudo-inverse method uses singular value decomposition and gives a minimum-norm solution [16].

To compare analysis to numerical simulations and experiments, a method of calculating the expected statistics has to be developed. The covariance matrix of the estimated angular rates can be used to determine the analytical variance of the data [13]. The covariance matrix, R, is simply composed of the uncertainty associated with each sensor. Assuming that the standard deviation of the noise for each sensor measurement is uncorrelated, the covariance matrix for the two sensor system is given as

$$R = \begin{pmatrix} \sigma_1^{\,2} & 0 \\ 0 & \sigma_2^{\,2} \end{pmatrix} \qquad (2.7)$$

It is shown in [13] that the theoretical variance of the estimated rate can be computed as

$$Var(\hat{\omega}) = (H^T R^{-1} H)^{-1} \qquad (2.8)$$

13

The variance describes how far a set of numbers is spread out. A variance of zero would indicate that all the values are identical and the estimated gyro rates are not affected by noise. A small variance indicates that the data is very close to the mean or expected value, while a high variance indicates that the data is very spread out around the mean. An equally valuable measure is the square root of the variance, called the standard deviation, σ. The standard deviation has the same dimension as the data, and therefore is easy to use in comparisons of deviations from the mean.

To see what effect gyro noise has on the integrated signal, which is normally used to predict the attitude, the following analysis may be performed [17]. The rectangle rule is assumed to be used to perform the integration. Let N_i be the ith random variable in the white noise sequence. Each N_i is identically distributed with mean $E\{N_i\} = E\{N\} = \mu = 0$ and finite variance $Var(N_i) = Var(N) = \sigma^2$. The result of using the rectangular rule to integrate the white noise signal $V(t)$ over a timespan $t = n \cdot \delta t$ is

$$\int_0^t V(\tau)d\tau = \delta t \sum_{i=1}^n N_i \tag{2.9}$$

where n is the number of samples received from the device during the period, t, and δt is the time between successive samples. Since $t = n \cdot \delta t$ and $Var(N) = \sigma^2$, it follows that the variance is shown in (2.10) [17]:

$$Var(\int_0^t V(\tau)d\tau) = \delta t^2 \cdot n \cdot Var(N) = \delta t \cdot t \cdot \sigma^2 \tag{2.10}$$

As shown in Figure 6, noise introduces a zero-mean random walk error into the integrated signal, whose standard deviation (2.11) is proportional to the square root of time [17].

$$\sigma_\theta(t) = \sigma\sqrt{\delta t \cdot t} \tag{2.11}$$

Note that (2.10) is simply the square root of (2.11) with $\sigma_\theta^2 = Var(\int_0^t V(\tau)d\tau)$.
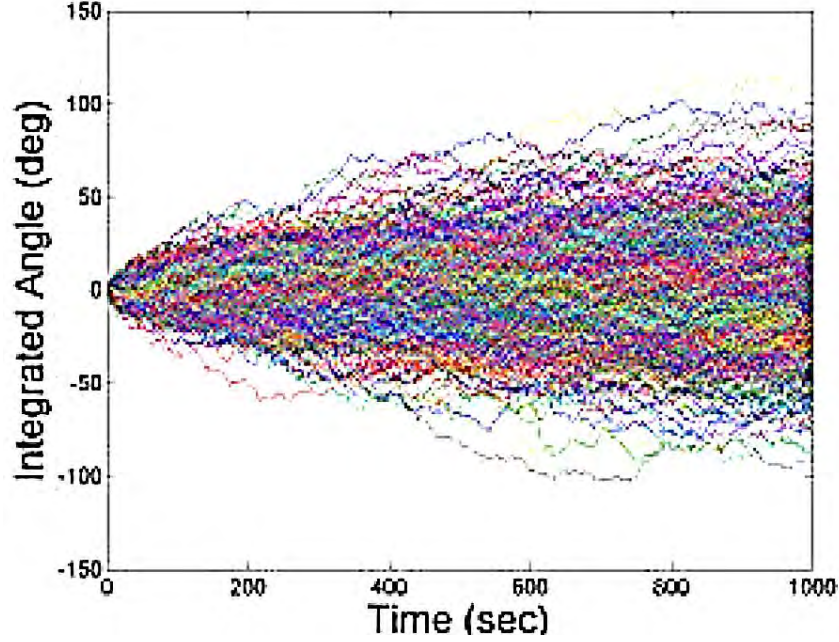
Figure 6   Zero mean random walk error from integrating noise of a rate
sensor,1000 trails, from [18]

Industry is typically interested in how the noise affects the integrated signal, so gyro manufactures commonly specify the noise using the angle random walk (ARW) measurement [17], which depends on the variance of the integrated noise.

$$ARW = \sigma_\theta(t)(^\circ / \sqrt{h})$$ (2.12)

The ARW has units of degrees per square root hour. A typical MEMS gyro has an ARW of 0.2 $(^\circ / \sqrt{h})$. This means that after one hour, the standard deviation of the orientation error will be 0.2°, after two hours, it will be $\sqrt{2} \cdot 0.2 = 0.28°$, etc. Clearly, the ARW is a function of the gyro noise so ARW can be reduced by improving the precision of gyro measurements.

In order to develop data that can be analyzed numerically, MATLAB was used to create and simulate each sensor and its corresponding measurements. The elevation angles EL^1 and EL^2 for the initial test were chosen randomly to be 80° and 45°, respectively. The estimated angular rate for each sensor measurement is found using (2.6). This calculation is repeated 10,000 times, each time with a random value for

15

the sensor noise V. The array of estimated angular rates can now be analyzed to find the relevant statistics using the following equations [19]:

$$Mean(\mu) = E\{x\} = \frac{1}{N}\sum_{i=1}^{i=N} x_i$$

$$STD(\sigma^2) = E\{x^2\} = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}(x_i - \mu)^2}$$

$$Skew(\gamma) = E\{x^3\} = \frac{\sum_{i=1}^{i=N}(x_i - \bar{x})^3}{(n-1)\sigma^3}$$

$$Kurtosis(\kappa) = E\{x^4\} = \frac{\sum_{i=1}^{i=N}(x_i - \bar{x})^4}{(n-1)\sigma^4}$$

(2.13)

One of the most descriptive statistics and perhaps most important measure of central tendency is the first statistical moment, $E\{x\}$, or the mean. The mean, μ, is found by summing all the data, x_i, and dividing by the total number of data points, N.

The second statistical moment, $E\{x^2\}$, or the standard deviation, σ provides an idea of how far the data is spread out from its mean. Along with variance, σ^2, this statistical property is among the most familiar and useful within the category of measures of dispersion. The standard deviation is defined as the square root of the average squared distance of each datum from the mean.

The third statistical moment, $E\{x^3\}$, or skewness, γ, gives an indication of the asymmetry of the data. If the data is perfectly symmetrical about the mean, then γ will be 0, as it is for a normal distribution. If γ is negative, then the left tail of the distribution is longer than the right. If it is positive, then the opposite is true.

The fourth statistical moment, $E\{x^4\}$, or kurtosis, κ, is an indication of the pointedness of the data's distribution. If κ is large then most of the standard deviation is caused by extreme deviations from the mean. If κ is small then most deviations are nearer the mean and the distribution is rounded.

All four statistical moments will be used to evaluate each individual sensors signal, as well as the combined output of the array. Figure 7 provides a visual comparison

16

between each simulated sensor output along the y-axis and the output of the combined gyro array. Notice by visual inspection that the dispersion appears to have grown.
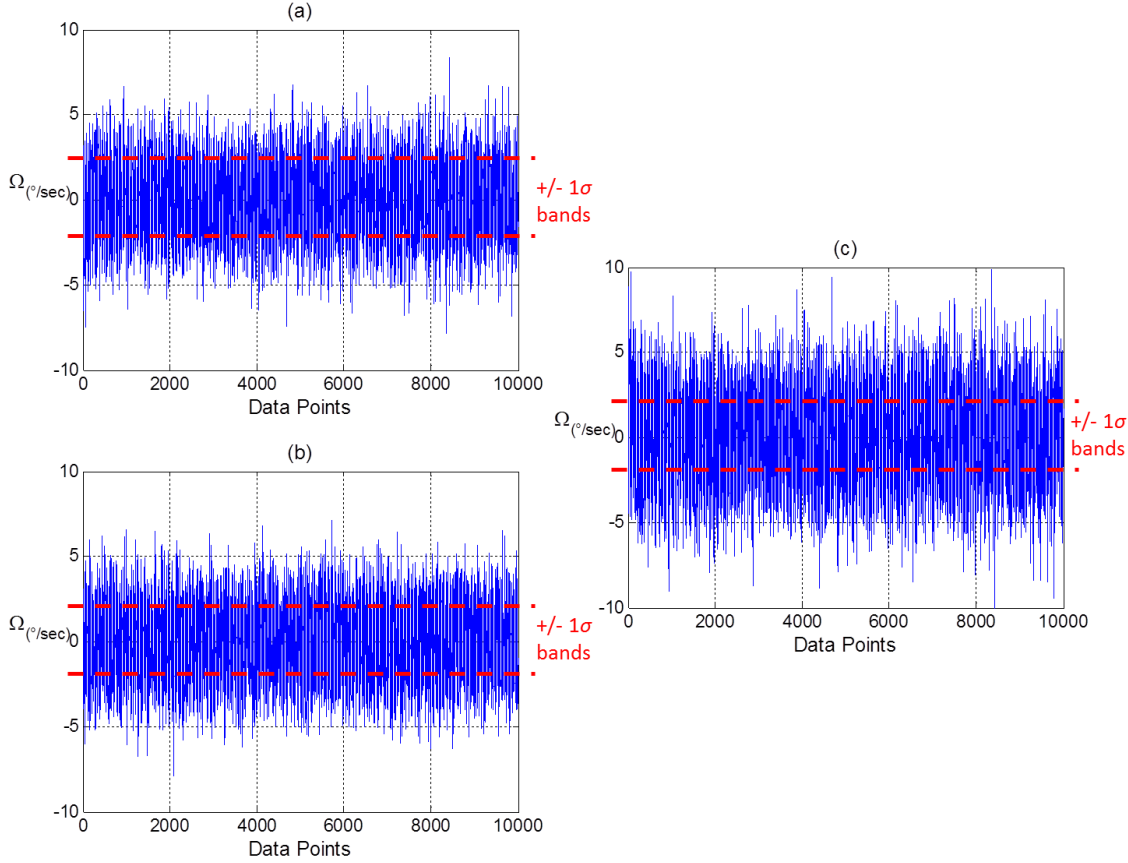
Figure 7   Measurement noise for: (a) sensor 1 with $EL^1=80°$; (b) sensor 2 with $EL^2=45°$; (c) combined array

A statistical analysis was performed on the data displayed in Figure 7 using equations (2.13) in order to compare results. The statistical results and a histogram with a normal distribution curve are displayed in Figure 8. The standard deviation of the combined result is, in fact, larger than that of the individual sensors. The increase in standard deviation translates into an increase in the ARW parameter. Consequently, the redundant sensor array would perform worse overall than if either of the sensors were used individually.
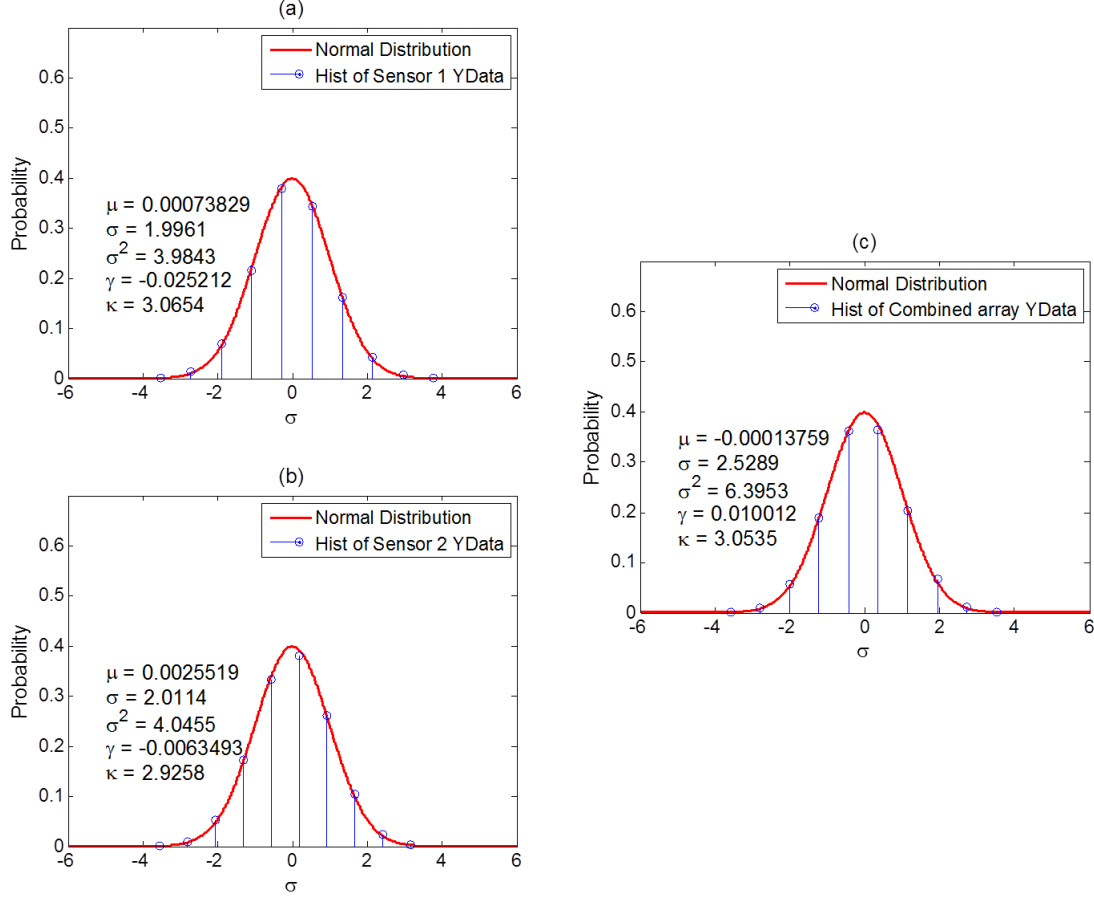
Figure 8　　Statistical results: (a) sensor 1; (b) sensor 2; (c) combined array with $EL^1=80°$, $EL^2=45°$

During the simulation process it was discovered that by varying the elevation angles, EL^1 and EL^2, the dispersion of the combined results varied dramatically. Therefore, it is necessary to find the ideal angles for configuring the array in order to get the best measurement precision. The next section investigates a procedure based on observations given in [13] for optimizing the geometry of the array.

C.　　IDEAL ARRAY CONFIGURATION

From the analysis above, a performance index (*PI*) by which to measure the efficacy of the sensor array is chosen. One such *PI* is the trace of the co-variance matrix [13]. The magnitude of the trace is an indicator of the error associated with the array. This is the criteria, or cost function, used to optimize the configuration. The trace

18

of the co-variance was chosen because it incorporated both the elevation angles and sensor noise into one simple equation. The PI for the two sensor array is given as

$$PI = trace(Var) = trace((H^T R^{-1} H)^{-1}) = \frac{\sigma_1^2 + \sigma_2^2}{\sin(EL^1 + EL^2)^2} \qquad (2.14)$$

The PI will be used to determine the elevation angles for each sensor that optimizes the geometrical configuration for the array. The expression for the optimal elevation angles is found by taking the partial derivative of the PI with respect to each elevation angle, then solving for the angles that minimize that function. The partial derivatives of the PI with respect to each elevation angle are identical:

$$\frac{\partial PI}{\partial EL^i} = -\frac{2\cos(EL^1 + EL^2) * (\sigma_1^2 + \sigma_2^2)}{\sin(EL^1 + EL^2)^3} \qquad (2.15)$$

From (2.15), the resulting EL^1 and EL^2 that minimize the PI are any combination of EL^1 and EL^2 that give an orthogonal arrangement of two sensors.

$$EL^1 = 90° - EL^2$$
$$EL^2 = 90° - EL^1 \qquad (2.16)$$

The analysis above was verified numerically using Microsoft Excel to minimize the trace of the variance matrix, and graphically by using MATLAB to create a three dimensional surface plot for visualization. The statistics are compared next, to check the validity of the optimization results.

D.   VERIFICATION AND VALIDATION OF RESULTS

For a given $\sigma_1 = \sigma_2 = 2$, the value of the performance index is $PI = 8$. The two angles calculated using a Microsoft Excel solver varied depending on the initial start values of EL^1 and EL^2. However, the elevation angles agreed with expression (2.16) in that they always resulted in a separation of 90 degrees. To verify that the calculated angles did indeed minimize the variance, MATLAB was used to develop the arrays of estimated angular rates ($\hat{\omega}_{1,2}$). The arrays were then analyzed to compute the actual variance. The results are displayed graphically in Figure 9 in order to provide a visual comparison between the outputs of the optimal and non-optimal configuration. The text box in the

lower left of each figure displays the *PI* calculated from (2.14). Note again that these results are for identical sensors ($\sigma_1 = \sigma_2$).

$PI = 8$ $PI = 24.3$
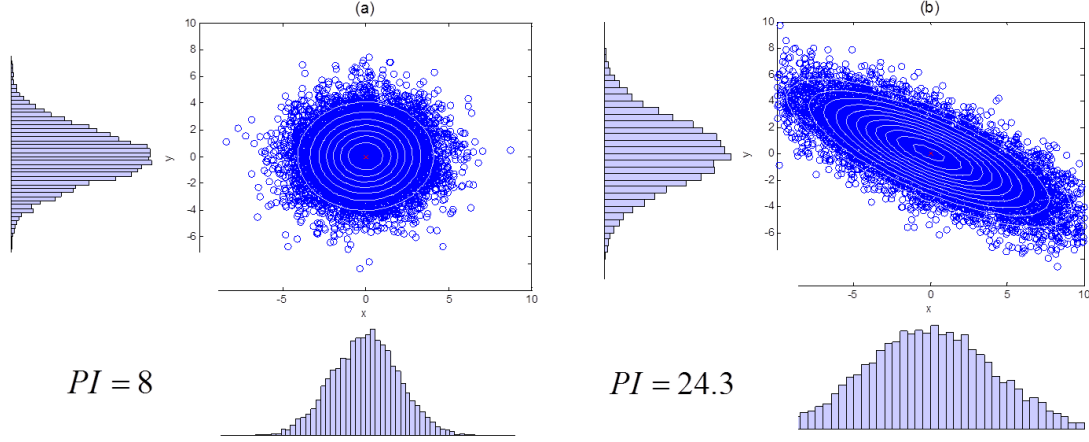
Figure 9   Array noise outputs: (a) optimal configuration with $EL^1=30°$ and $EL^2=120°$; (b) non-optimal configuration with $EL^1=80°$ and $EL^2=45°$

To ensure the optimal configuration was indeed an improvement over the non-optimal case the same statistical analysis was conducted for the case where $EL^1=30°$ and $EL^2=120°$. In the optimal case, the difference between the angles is equal to 90°. Figure 10 provides the same comparison as Figure 7, except this time the array is set to an optimal geometry. By visual inspection, it is apparent that the precision is increased over the non-optimal case of $EL^1=80°$ and $EL^2=45°$ (see Figure 7).
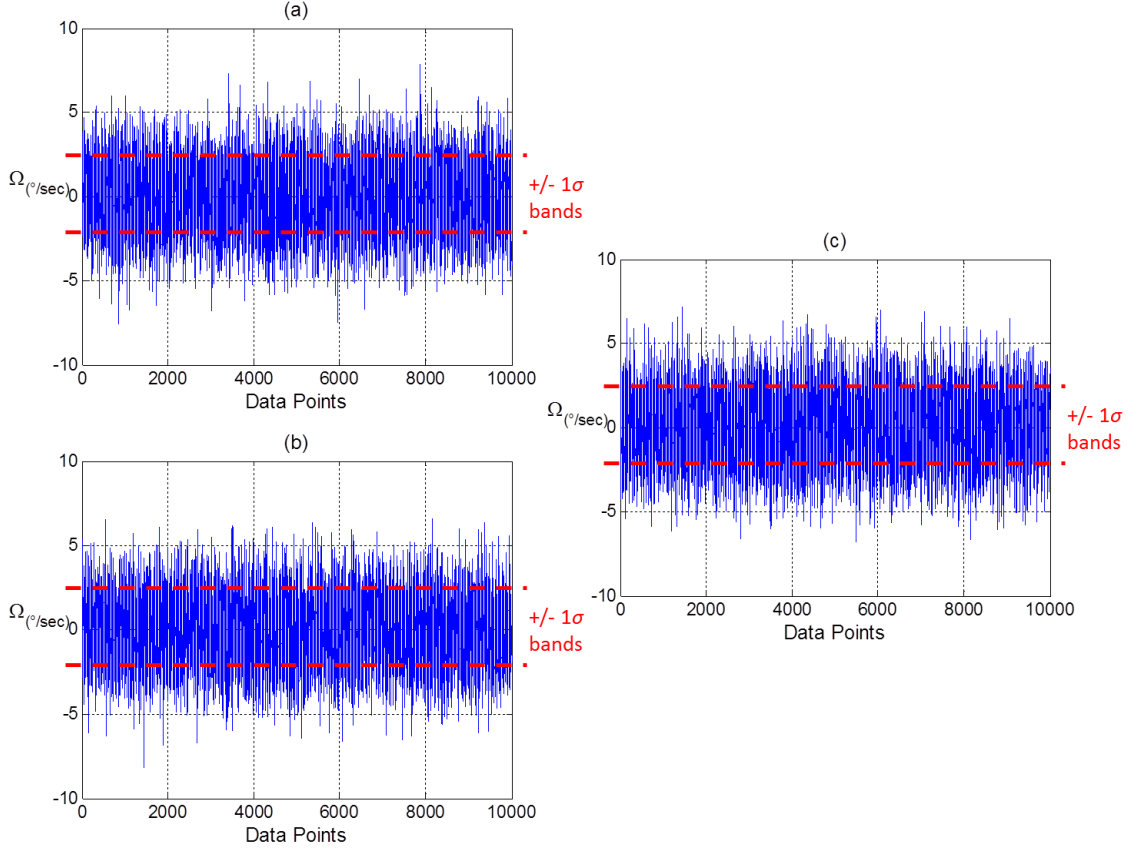
Figure 10   Measurement noise for: (a) sensor 1 with EL^1=30º; (b) sensor 2 with EL^2=120º; (c) combined array

In order to statistically compare results, analysis was performed on the data displayed in Figure 10 using equation group (2.13). The statistical results and a histogram with a normal distribution curve are displayed in Figure 11.
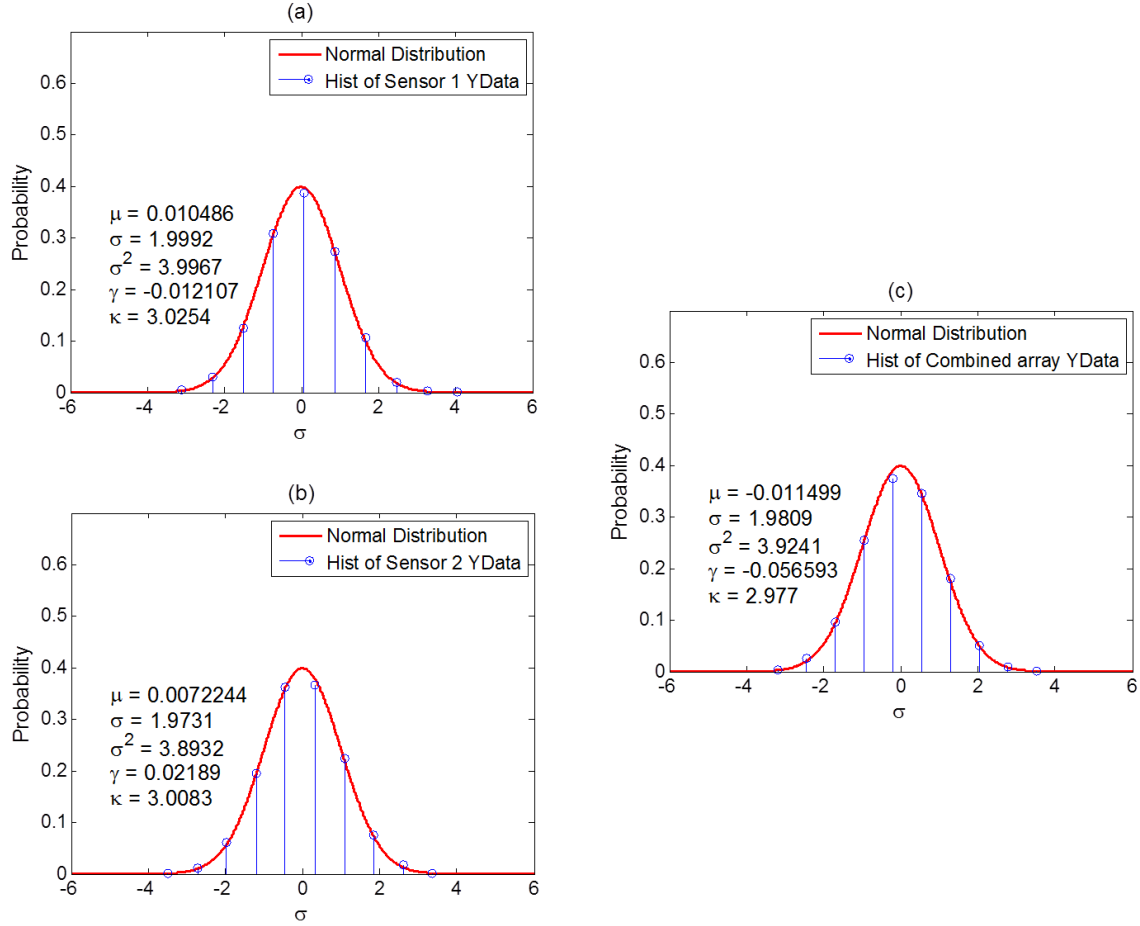
Figure 11   Statistical results: (a) sensor 1; (b) sensor 2; (c) combined array with
EL1=30º , EL2=120º

Notice the increase in improvement in the dispersion for the combined sensor array as compared to Figure 8. The standard deviation of the optimized geometry, (σ=1.98), is reduced as compared to the standard deviation of the non-optimized array (σ=2.53). This is a result that encourages additional research into the potential for optimizing the geometry of an array made of a large number of sensors. Notice the importance in choosing the correct or optimal configuration: If the incorrect elevation angles are chosen the precision of the combined array is actually worse than any of the individual sensors. It should also be noted that in the case of redundant sensors studied here, the precision of the two gyro array is the same as the precision of the best single

sensor. Thus, redundancy has been achieved with no improvement in precision. As will be seen later, improving the precision requires a carefully constructed three dimensional sensor array.

E.   SUMMARY

This chapter explained the theory behind improving the precision of a sensor array, and thus reducing the ARW of a rate sensor through the use of redundant sensors. Techniques and procedures from previous work were used to recreate results found in [13] and [17]. Statistical noise analysis was conducted to lay a foundation for the analysis that will following later chapters of this thesis. Finally, an investigation into optimizing the geometry of an array was introduced for two sensors configured in a single plane.

THIS PAGE INTENTIONALLY LEFT BLANK

III. CONSTRUCTING A SENSOR ARRAY TEST BED

A. INTRODUCTION

Before a complete system is designed, most of the individual components making up the design must be well-defined. The initial stages of the design process are used to gather this information, so that a complete architecture can be described. The first step is to gather a set of requirements, and then refine them into a specification that contains enough information to begin designing the system architecture. A large part of the work associated with this thesis is associated with the design and construction of a sensor array test bed. The test bed will be used to house the sensor array and all the supporting equipment so that multi sensor experiments can be easily conducted. The test bed must therefore, include any necessary computer processors and a power distribution sub-system. The design of this system followed a top-down methodology starting from a set of determined requirements. As shown in Figure 12, this chapter will discuss the design of the system from the top down. This top-down methodology is a process taught in most embedded systems texts and is commonly used in industry [20].
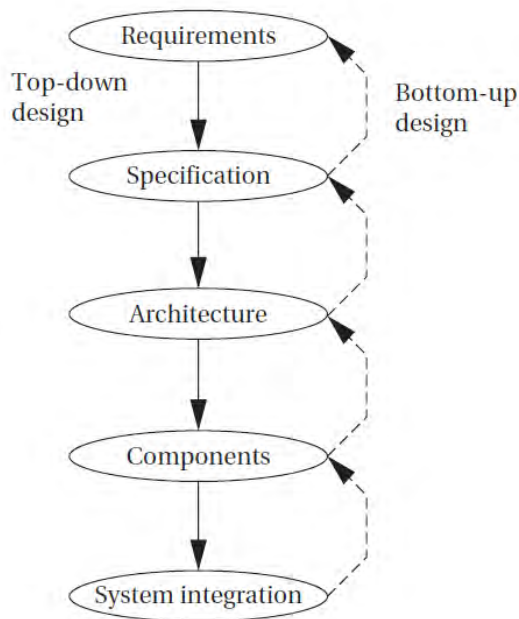
Figure 12   Levels of abstraction in the design process, from [20]

25

B. REQUIREMENTS

This section will focus on the functional requirements of this system. The basic functional requirements of the test bed can be summarized in to the sub categories listed below.

- Purpose: The purpose of the system is to acquire, process, and display real time data from an adjustable 12 sensor array.

- Inputs and outputs:
  1. Types of data: Analog electronic signals converted to digital data.

  2. Data characteristics: Calibrated raw digital data

  3. User inputs: Are required to program new software and mechanically manipulate the sensor orientations.

  4. Types of I/O devices: i) Universal serial bus or USB to RS232; ii) Onboard analog/digital converters

  5. Functions: User can program the on board chip to stream raw data to the processing controller. The controller can send unprocessed or processed data to the visual display on a host PC for collection and further analysis.

- Precision: Real-time processing is desired for the most accurate state estimations.

- Manufacturing cost: Fiscal environment consideration encourages efforts to minimize cost as much as possible.

- Power: Capable of running system on standalone 24VDC battery power supply.

- Physical size and weight: Desktop mounted structure that can be easily manipulated to support multiple sensors, a power supply, computer, and router. Ideal is a weight of approximately 5–10 lbs.

C. SPECIFICATIONS

The specifications level is the next level down from requirements in the flow of design. When compared to the requirements, the specifications are more precise. The specification of the system is typically more carefully written so that it accurately reflects the requirements in a way that can be clearly followed during the design phase [20].

The major function of the system includes a test bed structure that will facilitate multiple array configurations while providing structural support for the computer, the power distribution system, and 12 or more sensors. The system was designed using a top down process starting with the Sparkfun razor 9 degrees of freedom (9DOF) IMU [21], as the sensing element of choice. The razor was the first component selected, the rest of the test bed components were constructed to support it.

The Sparkfun Razor was selected because it is a low price commercial off the shelf system with a history of dependability. The Razor IMU has three independent sensors on a single board that can be used to increase the amount of research that can be done using the test bed. The Razor also uses a standardized common interface to each sensor onboard, making the overall system easy to integrate and use. Each board ties its three sensors to the central ATmega 328 onboard processor, which combines the data into a single stream.

The rest of the test bed needs to be designed around the specific needs of the Razor IMU. The requirements are:

- Inputs and outputs: Digital data in and out through serial interface
- Data characteristics: Calibrated raw digital data in ASCII string format with eight bits per data element
- Occasional user inputs: Required to flash the onboard chip with new software
- Types of I/O devices: USB to RS232
- Functions: User can program the on board chip to stream raw data to the processing controller
- Power: Requires 3.5-16VDC power
- Support structure: Structure to mount each sensor board

To increase the number of possible sensor configurations the support structure should have as many sides or mountable surfaces as possible.

Taking advantage of the common power requirements of all the components simplifies the electrical power system (EPS) design. The EPS should therefore supply a nominal 12VDC for all of the components. The sensor will use its onboard voltage

regulator to drop the 12V supply down to the required 3.3VDC. Other devices can also regulate voltage similarly or additional power conditioning can be added, as necessary.

D.   TEST BED ARCHITECTURE

The architecture level is the next level down from specification level in the flow of design. The specification level does not describe how the system does things; it only provides the description of what the system does. Describing how the system implements these functions is the purpose of the architecture step in the design process. The architecture is a plan for the overall structure of the system that will be used later to select or design the components that make up the test bed. To understand an architectural description of this test bed, a system level block diagram is shown in Figure 13. The system level architecture description is in the form of a block diagram and shows the major operations and how data flows among them. This block diagram is intended to be abstract. It does not specify what operations will be performed by software running on a CPU, or what will be done by specific pieces of hardware.
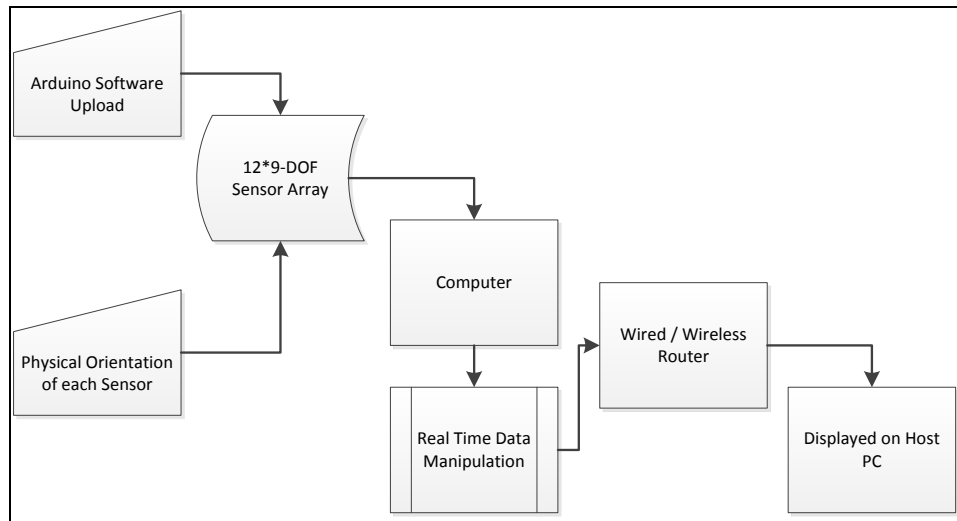
Figure 13   System level block diagram for the test bed

Once each sensor's onboard processor, (ATmega328), is loaded with firmware and hooked up to a DC power supply, the IMU can incorporate data from each of its three sensors, a MEMS triple-axis gyro, a triple-axis accelerometer, and a triple-axis

28

magnetometer. These nine degrees of inertial measurement data are then streamed into a serial interface. In order to acquire, process, and display the data, a computer with real time processing capability is required. National Instruments produces the cRIO-9024, embedded real-time controller, that features an industrial real-time processor for deterministic, reliable real-time data acquisition and management [22]. This computer was chosen for data acquisition based on the fact that it meets the serial interface and real-time data acquisition / data management requirements.

The computer and the sensors must be powered. However, the distribution system is fairly simple since the individual sensors, computer, and computer modules, as well as the router can all run on a common 12V DC power supply. When the test bed is powered up the sensors will start streaming their 9DOF data into the cRIO real-time processor. From the cRIO, the data can be routed to the host PC for further analysis and display. Now that an architecture for powering and streaming data has been developed, it is time to focus on the support structure requirements. The support structure needs to be sturdy enough to provide support for the computer, power distribution system, and router and serve as an adequate test bed that can host multiple sensor array configurations. The structure must also be stiff enough to prevent flexible effects from corrupting the measurements.

Initially a truncated icosahedron [23], was considered as the main support structure. The truncated icosahedron is a sturdy structure and provides 36 unique surfaces that could be used for mounting sensor packs. The structure is made up of different size panels that, when 3-D printed, have to be connected at various angles. Because of this disadvantage a regular icosahedron, shown in Figure 14, was constructed instead. The regular icosahedron proved to be much easier to design and manufacture, while retaining most of the desirable characteristics of the truncated icosahedron. A frame with this shape can be created from 20 identical equilateral triangles. This greatly simplifies the manufacturing process, but can limit the number of possible array configurations. Each of the 20 faces has the potential to hold one of the 12 sensor panels. To further expand the number of possible sensor array configurations each panel was designed to provide three possible sensor positions.
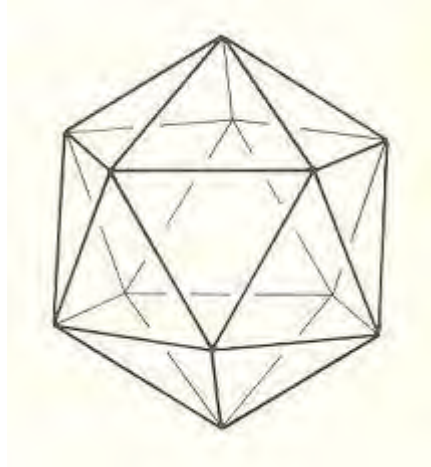
Figure 14   Regular icosahedron

E.   COMPONENT DESIGN AND SYSTEM INTEGRATION

The system integration and component selection are the last two levels in the design process. In this section, each component and how it fits into the test bed will be described.

1.   9DOF Sparkfun Razor

Each IMU has three separate sensors onboard. They are an accelerometer, magnetometer, and gyroscope. Many sensor characteristics were considered when analyzing the 9DOF IMU. Overall, it was decided that a single chip that hosted sensors with mid-level characteristics could be used to adequately perform the experiments. The 9DOF Sparkfun razor IMU met the requirements of this thesis. The sensors on any IMU can be characterized by their range, resolution, axes, power, and interface requirements.

The range of values that a device is capable of measuring is an important factor when deciding what sensor is appropriate for use. Obviously, a 24g accelerometer should not be used to track body motion. Likewise, if the accelerometer tops out at 1g then the sensor may not provide useful data on, say, a rocket launch for example. This same point can easily be made for the gyro and magnetometer. A sensor with the widest range possible will increase the number of tests that can be done. Therefore, a wide range sensor is the most desirable for use in this thesis. Unfortunately, increased range is

30

usually inversely proportional to resolution. Here, this is an advantage as the desire is to improve performance using redundant measurements.

The best way to understand the concept of range and resolution is by comparison with a yardstick. For example, when a 1 meter yardstick is divided into millimeters the resolution is 1/1000. The range of the yardstick is 1000 millimeters. The smallest "tick" on the yardstick is the resolution. The Sparkfun razor incorporates an analog to digital converter (ADC). The resolution of the onboard 16-bit ADC is a function of how many parts the maximum signal can be divided into. An ADC takes an analog signal and turns it into a binary number. Thus, each binary number from the ADC represents a certain voltage level. Resolution is the smallest input voltage change a digitizer can capture. Resolution can be expressed in terms of the number of bits, or as a proportion, or as the percent of full scale. The formula to calculate range of an ADC is 2^n. For example, the gyro on board the Sparkfun IMU uses a 16 bit ADC with an additional bit for sign. Therefore, it uses $2^{16} = 65,536$ bits hence the resolution is 1 part out of 65,536. For the given range of + or − 2000 (°/sec), that represents a sensitivity of 0.069 (°/sec). Resolution is the most important characteristic to consider when analyzing the accuracy on individual IMUs or an entire array. Resolution limits the precision of a measurement. The higher the resolution the more precise the measurement will be [24].

The axes refer to the number of directions in which a quantity can be measured. Accelerometers measure acceleration along the specified axes, whereas, gyros measure acceleration around a single axis and magnetometers measure both the magnitude and direction of the earth's magnetic field in relation to a given axis. Each sensor has three measurement axes, for a total of nine axes of measurement on the IMU. The integration software included on the sensors Arduino board incorporates data from all nine axes to determine an individual sensor's yaw, pitch, and roll. In order to accomplish this, the board manipulates raw digital data from the output of each sensor. The research in this thesis only focuses on the raw data from the gyros, with the goal to increase the precision of an array. Further processing to estimate attitude angle is a logical extension, but is beyond the scope of this work.

The power requirement of a sensor represents the amount of power that the device will typically consume during operation. It is important to consider the power requirements for all sensors and any other devices that will be a part of the system. The requirements for each sensor and all the related support devices factor into the distribution system design. All the sensors on board the 9DOF IMU receive regulated voltage from the onboard regulator. This simplifies the power system design, and thus the overall test bed design. The details of the electrical power system will be covered later in the EPS construction section.

The method by which data is sent and received between a controller and a device is called the interface. There are several standards available and each has its advantages and disadvantages. Analog signals are easy to read and can be measured by most microcontrollers with very little code. Serial or I2C are common in situations where multiple pieces of information need to be read out to a controller. I2C is a two-wire serial interface that allows several devices to share a bus and communicate with each other. The 9DOF Razor utilizes a common I2C bus to communicate with the individual sensors. This type of interface is the most advantageous for gathering and sending multiple simple data signals at once. This allows the microcontroller to receive all valuable elements of a data string for a given time interval. Currently, most measurement devices and microcontrollers utilize the I2C serial interface for this reason. Once collected by the processor the data is repackaged and output to a serial bus so it can be read by the user.

The accelerometer on board the 9DOF razor is the ADXL345, a mid-range accelerometer with high resolution (13 bits) measurement up to 16g's. The ADXL345 measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/bit) enables measurement of inclination changes less than 1.0°. The ADXL345 is supplied in a small, thin, 3 mm × 5 mm × 1 mm, 14-lead, plastic package [25].

The magnetometer used by the 9DOF razor is the HMC5883L. It is a triple-axis, digital magnetometer developed by Honeywell, with a surface-mount, multi-chip module, designed for low-field magnetic sensing with a digital interface for applications. The HMC5883L includes a 12-bit ADC that enables 1° to 2° compass heading accuracy. The

HMC5883L is manufactured in a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include mobile phones, netbooks, consumer electronics, auto navigation systems, and personal navigation devices [26].

The three-axis MEMS gyroscope, on board the 9DOF razor, has 16-bit ADCs and uses signal conditioning to minimize the high frequency noise introduced in its channels. The sensor consists of three independent vibratory MEMS gyroscopes which detect rotational rate about the X (roll), Y (pitch), and Z (yaw) axes. When the gyros are rotated about any of the sense axes, the Coriolis effect (described in Chapter I) causes a deflection that is detected by a capacitive sensing circuit. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using the on-chip 16-bit ADC previously explained. The full-scale range of the gyro sensors is preset to ±2000 degrees per second (°/s) with a resolution of 0.069°/sec. The ITG-3200 communicates to a system processor using the I2C serial interface [6]. The assembled razor IMU ships from the manufacture as shown in Figure 15.
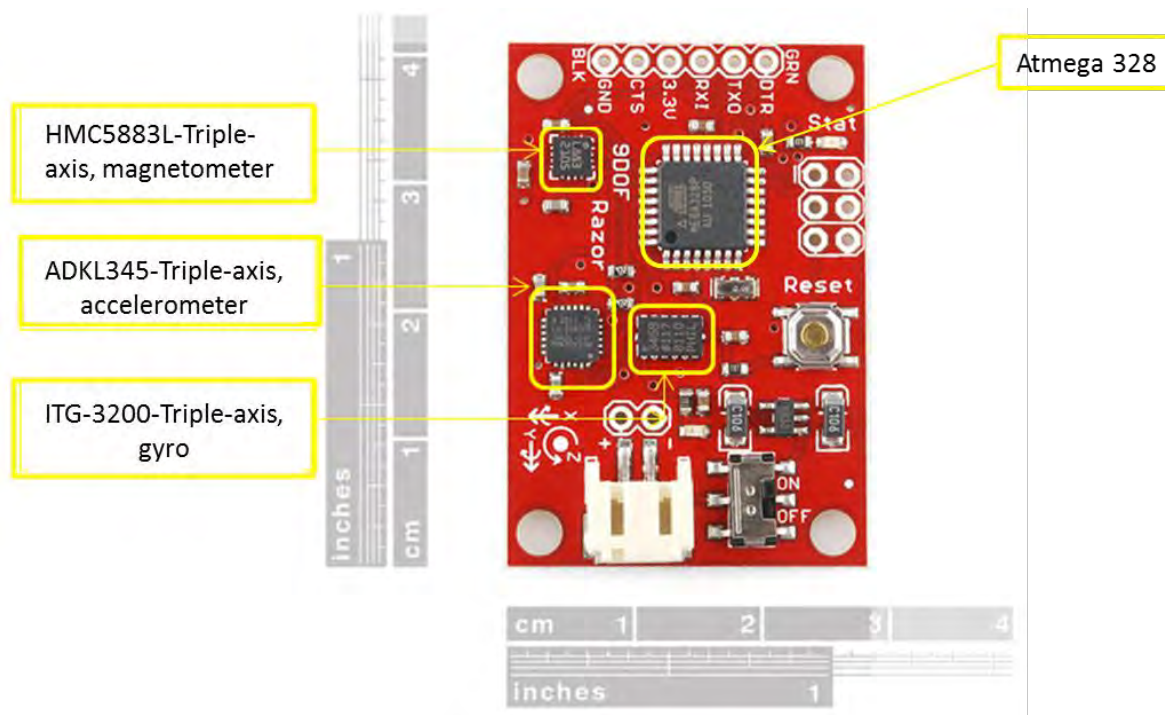
Figure 15   Sparkfun 9DOF razor IMU, after [21]

33

When connected to the computer the sensor is required to be wired to a FTS232 serial breakout board shown in Figure 16. This board is designed to convert RS232 to TTL and vice versa (TX and RX). This allows the microcontroller to communicate with a computer (e.g., the cRIO) that has a standard RS232 port. Once power is applied to the board, the level shifter converts CMOS TTL to RS232 and allows the computer to acquire the sensor data.
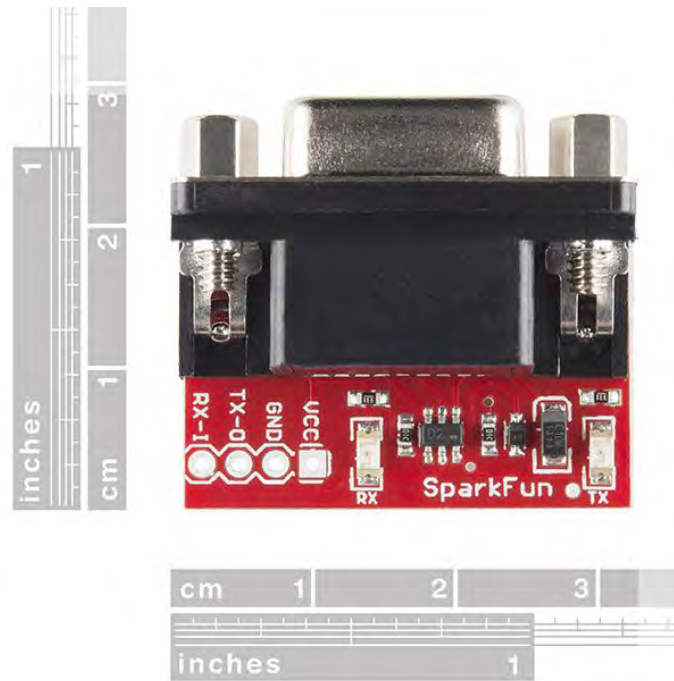
Figure 16   FTS232 serial breakout board, from [21]

2.   NI cRIO-9024 / 9113 Chassis and 9870 Modules

The NI cRIO-9024, embedded real-time controller is a programmable automation controller. It features an industrial 800 MHz real-time processor for deterministic and reliable, real-time applications [22]. This controller is designed for low-power consumption with dual 9 to 35 VDC supply inputs that deliver isolated power to the cRIO chassis and modules. The cRIO accepts a 9 to 35 VDC power supply and can function for long periods of time in remote applications using a battery [22]. The controller, shown in

Figure 17, provides two Ethernet ports—10/100 and 10/100/1000—that facilitate both the controller programming and communications to host applications, such as a PC for data visualization.

Figure 17   NI cRIO-9024 real-time controller, from [22]

The NI cRIO-9113 four-slot, reconfigurable chassis provides low-level hardware access to the NI 9870 module. The chassis allows for precise timing, triggering, control, and synchronization schemes for data acquisition applications. In order to acquire the data from the razor IMU, the chassis was configured to hold three NI9870 modules. Each of the NI 9870 serial modules adds four RS232 serial ports to the cRIO system. This provides up to 12 RS232 ports for interfacing with individual sensor boards. Using the LabVIEW FPGA API, the ports are accessed directly from the cRIO field-programmable gate array (FPGA). This NI-9870 module supports standard start bit, stop bit, and handshaking settings, and has baud rates up to 921.6 kb/s per port [22]. To communicate with the 9DOF sensor, the baud rate is set to 56.700 kb/sec. The NI 9870 uses an external power to provide maximum compatibility and reliability under all serial port conditions. A micro-fit pigtail power cable is connected to the 12V line of the EPS. The NI 9870 module and the cRIO-9113 chassis into which each module mounts are shown in Figure 18.

Figure 18   NI 9870 4-Port, RS232 serial interface module and the four-slot,
cRIO NI-9113 chassis, from [22]

All of the National Instrument components combine together to provide the computing power for the data acquisition system. After the supporting computer system was assembled, the test bed support structure was designed around it.

3.   LabVIEW

The cRIO is used to integrate the data streaming from the IMUs with a software program created by National Instruments. Laboratory Virtual Instrument Engineering Workbench, or LabVIEW for short, is a graphical programming environment commonly used in engineering [27]. LabVIEW relies completely on a graphical interface where nodes are wired together. LabVIEW compiles and checks the graphical code in real time. LabVIEW nodes can perform various tasks or calculations such as enumeration, sorting or simple mathematics. A piece of LabVIEW code looks similar to a block diagram or flow chart. At the most basic level, the programmer constructs a user interface to provide for data input and output, much like instrument panels on laboratory equipment. Hence, the written program acts like an instrument and is called a virtual instrument (VI) [27].

Often, a VI contains calls to subVIs. The subVIs are simply VIs that do tasks for the parent VI. This is similar to a traditional C++ program calling on functions. Multiple subVIs are used in the LabVIEW program designed for this thesis. The block diagram of the main VI is overly complicated, so it is not shown here. However, Figure 19 shows the user interface or front panel for the main VI. The main VI is used to monitor and collect the data streaming from each IMU. This VI was programed to allow the user to set the

serial port parameters and view the output for all ports. The serial port parameter selection area is highlighted in the red box. The parameters include baud rate, data size, and parity. The serial parameters are set to match the parameters set by the code that is flashed on each onboard processor. All chips are programed the same, with a baud rate of 57600 bits/sec, a parity of 1, and 8 byte data size. The individual data streams from each sensor are highlighted in the yellow box.

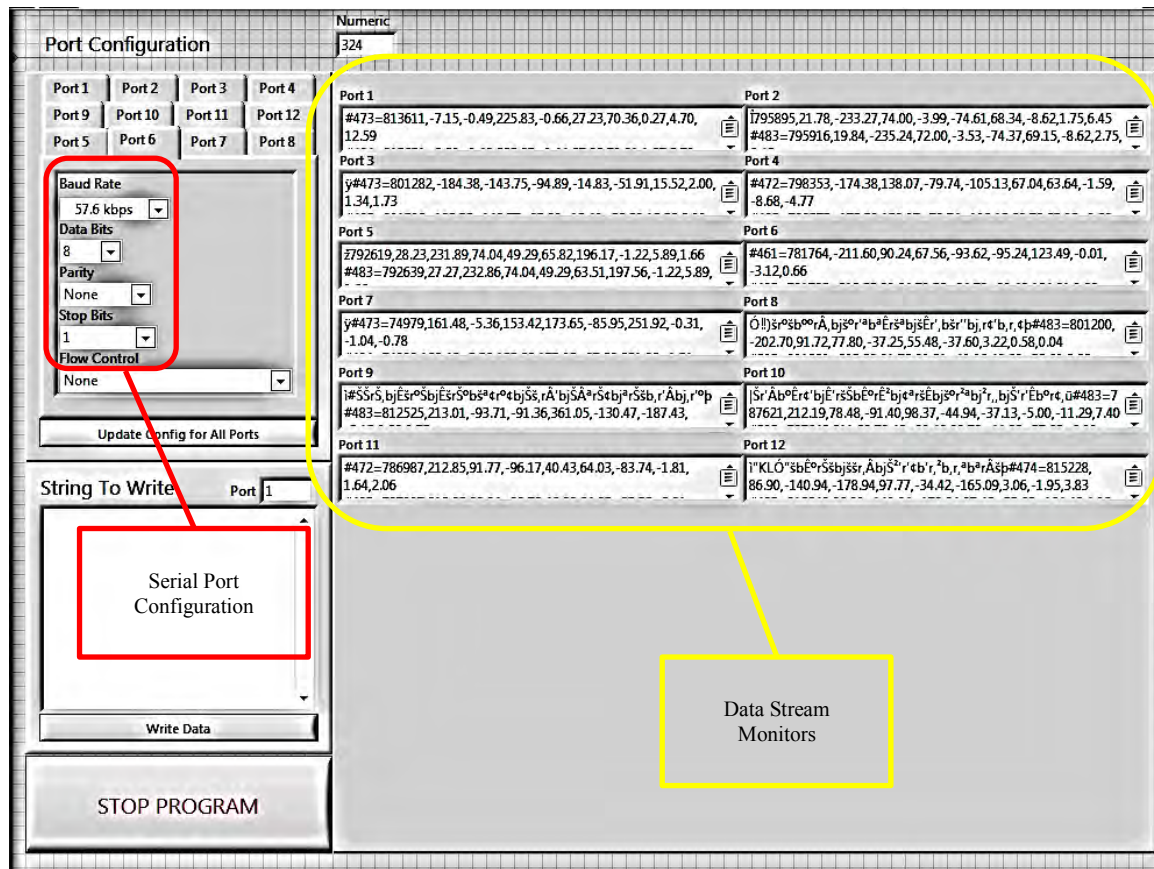Figure 19   Developed virtual instrument front panel for multi-sensor array

Hiding behind each front panel, is the working code or block diagram of the VI. This code is not shown due to its complexity. The code runs a producer and consumer loop that acquires data from the FPGA and routes it to the main VI for display. The main VI is able to sample the data directly from the FPGA and save it with minimal latency.

This is important because the data from each sensor is now being sampled at the same interval it is being produced. Thus the data can be correctly combined to provide an accurate measurement solution.

4. Icosahedron Support Structure

The test bed support structure was designed using the NX Ideas software package from Siemens [28]. There are many software packages available, NX Ideas was chosen because its simplicity minimized the time required to design the test bed. NX Ideas is an integrated product design solution that streamlines and accelerates the product development process for engineers who need to deliver innovative products in a collaborative environment [28]. The NX Ideas software offers additional applications than just the computer assisted design (CAD) tool that was used to design the structure. The 3-D printing files, called ".stl" files, can be exported after a completed design and then used by 3-D printers to create the object.

The NX Ideas design tool was used to virtually layout and design the test bed support structure. The main structure supports the sensors and all of the other necessary equipment, including the cRIO and power supply. As already discussed, an icosahedron was chosen to be the main support structure for the test bed. The size of each triangle was designed to allow a sensor panel to be mounted, and to allow the cRIO and all other components to fit inside the structure. Figure 20 shows the NX assembly of the 20 triangle forms into the completed icosahedron next to the manufactured assembly.
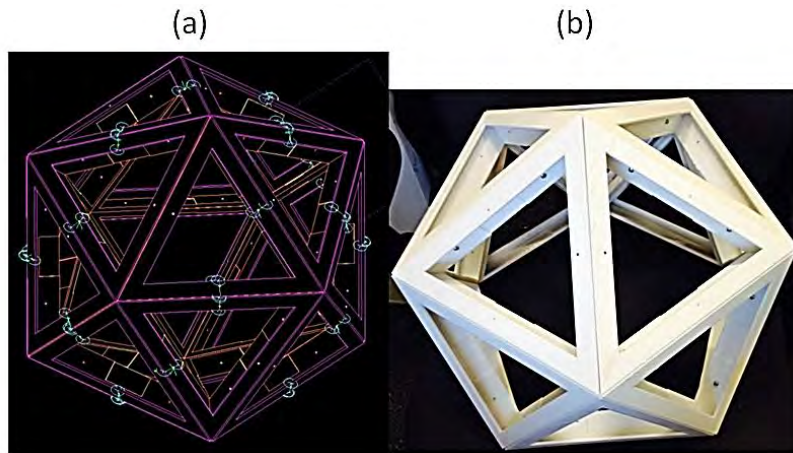
Figure 20   Sensor array structure: (a) NX model; (b) manufactured assembly

The completed icosahedron supports the sensor panels on the outside of each face. To house the other components of the test bed, an internal support structure was created. A simple shelf design is shown in the lower half of the icosahedron in Figure 21.
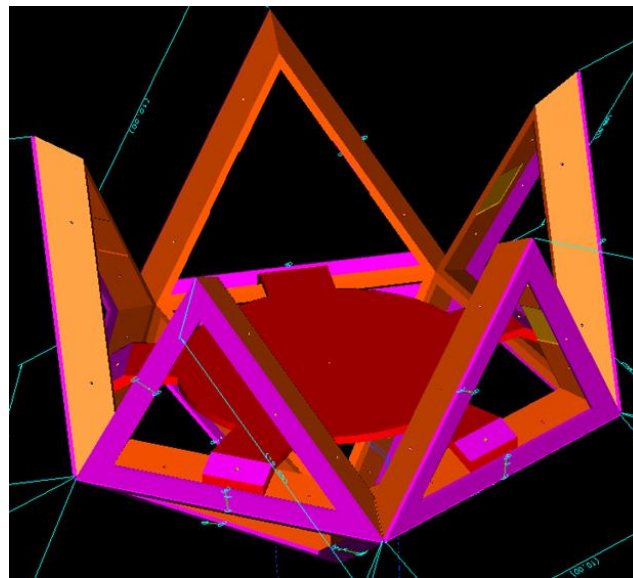
Figure 21   Lower half of icosahedron and the internal support shelf

In addition to designing the support shelf, a mounting harness for the cRIO and the battery were also created using NX Ideas and manufactured using a 3-D printer. The mounting harness for the cRIO and the battery are shown in Figure 22.
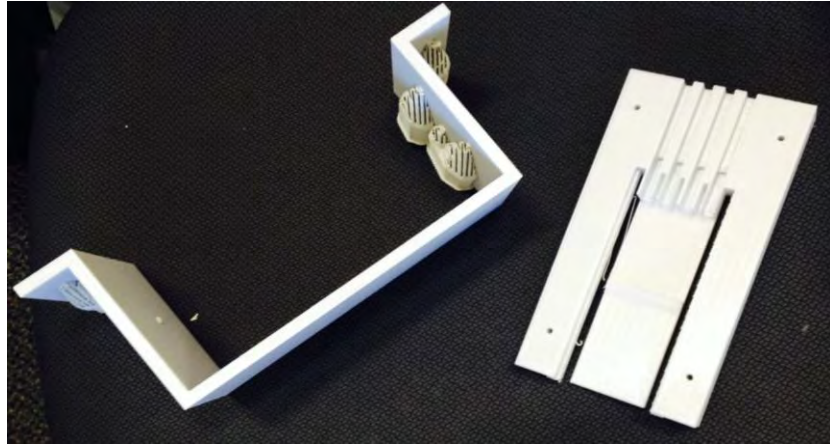
Figure 22   Constructed cRIO mounting harness (left) and battery module (right)

In addition to removing any expendable material necessary to provide support during manufacturing, all the pre-placed holes needed to be tapped for use with 4-40 machine screws. The construction was simplified using the same size holes and taps throughout the structure. A close up of the assembled computer mounting harness with the battery module is shown in Figure 23.
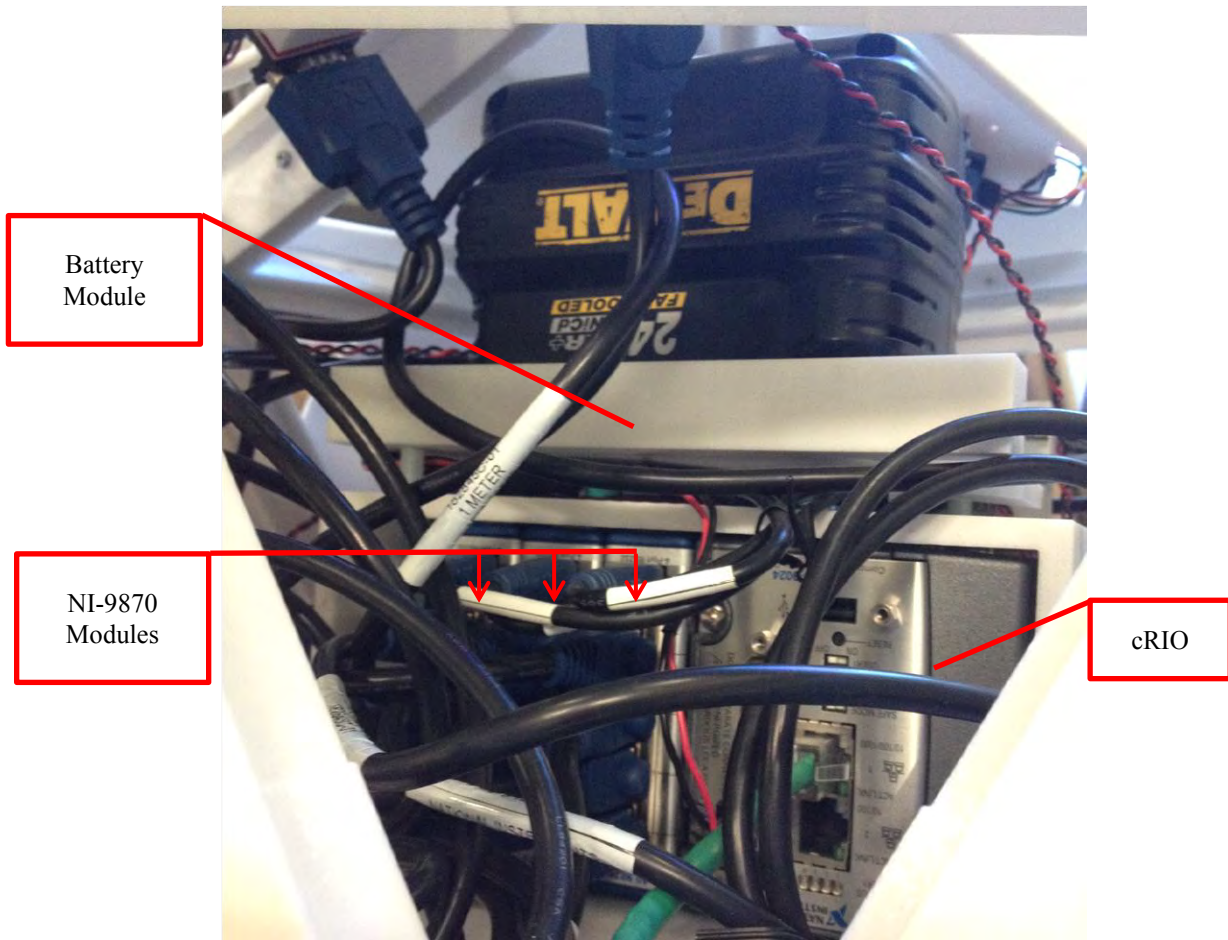
Figure 23   Mounting harness with cRIO and battery module as they are
assembled in the test bed

The individual sensor panels were also designed using NX Ideas and produced using 3-D printing. The panels need to support each sensor, the RS232 converter, and must have the ability to easily change the configuration of the mounted sensor. The panel is mounted to the test bed with attached sensor and RS232, shown in Figure 24. The three mounting holes are placed on the same base plate so that the sensor panel can be installed in any one of three orientations. This extends the number of possible sensor configurations on the test bed.
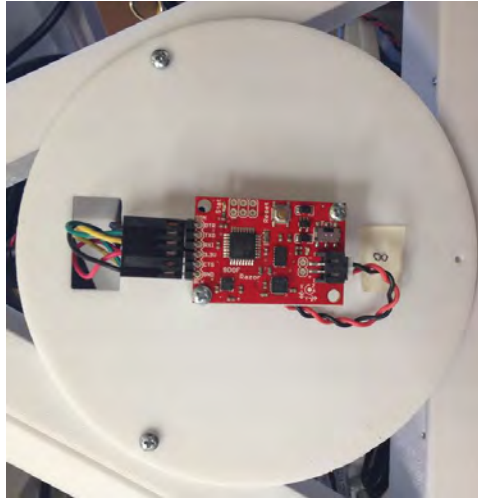
Figure 24　　Sensor panel with attached sensor

The completed test bed with the major support equipment and all the sensor panels attached is shown in Figure 25.
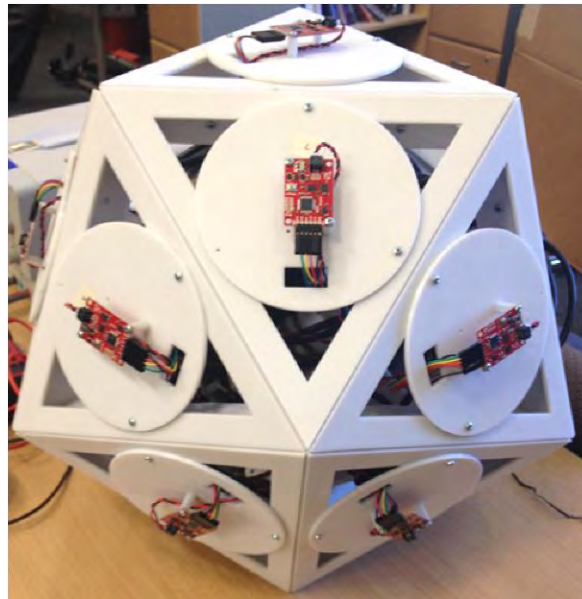
Figure 25　　Complete assembly of 3-D test bed

5.　　Electrical Power System

The design of the electrical power system (EPS) was simplified because all of the components on the test bed can run from 12 VDC. To increase the mobility of the test

bed, it was designed to use a 24 VDC battery. The power cell is a commercial De-Walt 24V cordless drill battery. The output of the battery is regulated down to 12VDC. The use of two 12V branch circuits allowed the total current load to be shared between two DC-DC regulators. The cRIO and its modules are connected on one 12V line, and all the sensors and wireless router are on the other 12V line. The flow chart in Figure 26 shows the EPS layout. To create a more modular system, all the components were connected to a distribution panel via pins. The only hard-wired connections are located on the back side of the distribution panels. The voltage regulators used were manufactured by Murata Power Solutions.
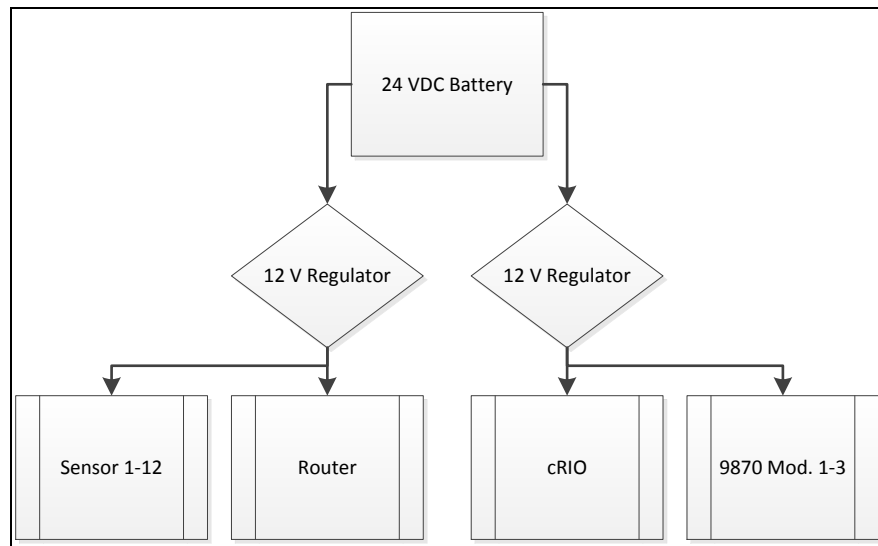
Figure 26   Electrical power system layout

F.   SUMMARY

This chapter described the requirements, specifications, and architecture of the designed multi-sensor test bed. The components selected and how they integrate into the system was also explained. The NX Ideas software and manufacturing tools used to create the icosahedron structure and mounting hardware were discussed. Using the LabVIEW program to interface the hardware and software was one of the most

challenging aspects of the entire process. None the less, it was possible to develop a real-time data acquisition system capable of collecting and storing data streams from 12 razor IMUs.

IV. INITIAL HARDWARE TESTING AND CALIBRATION

A. INTRODUCTION

This chapter presents the initial testing and calibration of the Sparkfun razor IMU. The Sparkfun razor is used to experimentally validate the analytical work performed in this thesis. The majority of the work performed during the initial gyro testing was focused on interfacing the gyros with the computer. Interfacing the hardware was necessary in order to characterize the statistics of the gyro signals. This chapter will conclude with a method for acquiring data and a statistical metric for qualifying it.

B. HARDWARE SETUP

Demonstrating the ability to properly code, program, and receive data from the IMU is important. The demonstration process is accomplished first with a single gyro then again with a three gyro array. This is done to work out any bugs before the ideas are scaled up.

For the initial validation tests, the serial four-pin output of a programed IMU is attached to a laptop that will store the data for later analysis. The preferred way in which a computer communicates with external devices is over the universal serial bus (USB). Most microcontrollers have USBs built-in, whereas others like the ATMega328 do not. The FT232R breakout board is required to translate the data transmitted over USB into something suitable for the serial universal asynchronous receiver / transmitter (UART) interface. The breakout board provides a micro-USB to USB connector on one side, and on the other, a six-pin serial layout [29]. The FT232R is shown connected to the Sparkfun IMU in Figure 27.
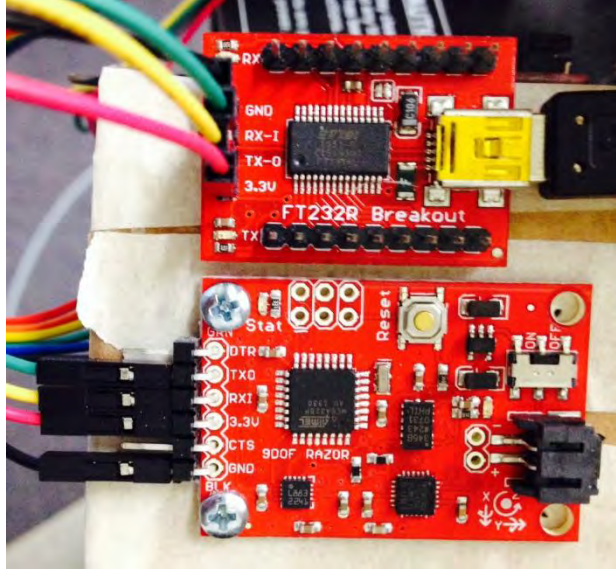
Figure 27   Sparkfun razor IMU (bottom) connected to the FT232R breakout
board (top) and then to a PC via a USB-micro-USB connector

Communication with the ATMega328 microcontroller requires a connection to the FT232R. The connection facilitates communication with the IMU, for example, loading a program onto memory, sending commands, receiving output data, and debugging the software program. The setup shown in Figure 27 is used to initially program, calibrate, and read data from the on board processor. The IMUs are pre-programed from vendor with firmware that displays a device self-check result and a menu. Upon initial power up the menu, shown in Figure 28, can be operated by the user to stream data from any of the sensors on board. The menu screen is observed via the serial port monitor in the Arduino program, which is used to compile the firmware and flash the ATMeg328.
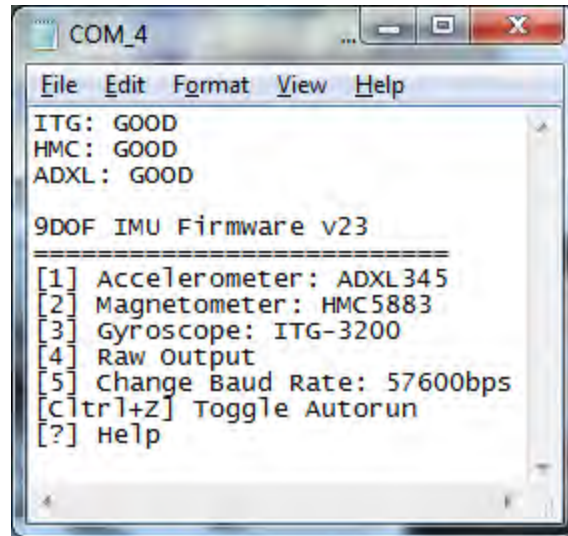
Figure 28   Arduino serial port monitor displaying the default menu screen for
the Sparkfun razor IMU

The firmware was modified, using the Arduino software, to skip the main menu and allow the device to stream its data directly after boot up, without input from the user. This allows for a more streamlined data acquisition process that proved helpful when using multiple sensors.

The firmware is also adjusted during the calibration process. The sensors arrive initially calibrated from the vendor. However, before using them calibration is recommended. For the most accurate results, the sensors should be calibrated in the same environment that they will be tested in. The precision and responsiveness of the sensor can be improved dramatically through calibration. If the sensors are not correctly calibrated, bias drifts in any of the sensing axes may cause measurement errors up to 20 °/sec [21].

The Sparkfun firmware has a "calibration" mode designed for the user to calibrate the onboard accelerometer, magnetometer, and gyroscope. The calibration procedure is lengthy and detailed . The tutorial details the calibration process and sets the IMU up to output data.

To view and collect the sensor data a software terminal program is used. RealTerm is a terminal program specially designed for capturing, controlling and

47

debugging serial data streams [30]. RealTerm's ability to capture streaming data makes it a valuable tool in this application. RealTerm can simultaneously read from several USB ports on a PC. A 4-1 USB dongle was used to facilitate testing with multiple sensors.

Figure 29 shows the RealTerm data acquisition screen. RealTerm does not have the ability to sync up data streams or manipulate the real-time data. However, it does have the ability to save the outputs to separate text files for later analysis using Excel or MATLAB.
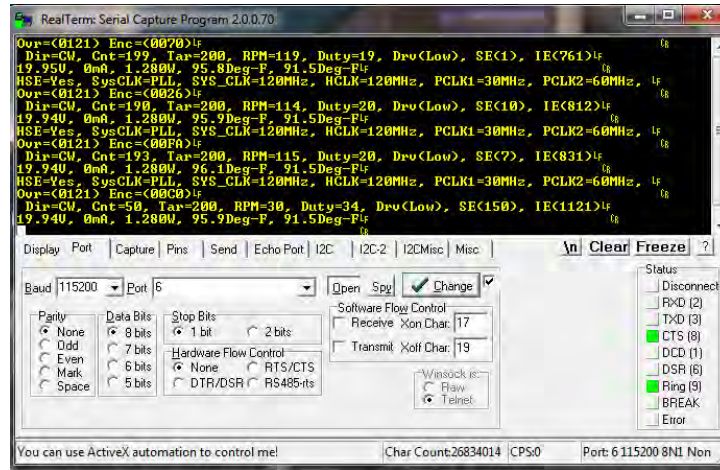
Figure 29   RealTerm serial data capture program

In order to verify that the gyros are operating correctly, the actual platforms position must be compared to the measured sensor outputs. The sensor was placed on the Torsion Control Plant (TCP) to compare actual position with measured position. The TCP is shown in Figure 30. The TCP is a precisely controllable platform that can support IMU testing by allowing the position of the platform to be driven by standard test signals [31].

Figure 30   Torsion control plant

C.   SINGLE SENSOR VALIDATION

The data from a single gyro was collected from the sensor array for the first validation experiment. The gyro on the IMU was aligned with its +z-axis pointing straight up. The gyro for this test is labeled S^3 in the test setup shown in Figure 31.
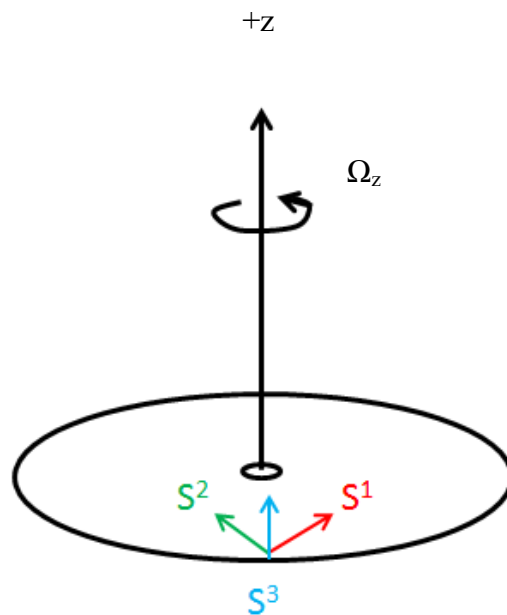
Figure 31   A three gyro array mounted on the TCP. The single sensor test was done with S^3 in blue. S^1 and S^2 used in later tests

The TCP was programed to conduct three consecutive 30º sine wave maneuvers. The TCP has an encoder that provides accurate position and time data. The encoder data allows for the comparison of the measured gyro data to the encoded data from the TCP. The data from both devices was analyzed in Microsoft Excel. The data was tabulated into columns and organized into rows with time stamps. Unfortunately, this setup prohibits the encoder and sensor data from being automatically synched up on a common time stamp. This is due the inability to start the clocks of the sensors at exactly the same time that the clock for the TCP is started.

The maneuver began with a few seconds of zero movement before completing three cycles of a 30° sine wave. The movement patterns were used to allow the encoder data to be manually synched with the measured data. Once this was accomplished the synched data could be exported to MATLAB for further analysis.

The TCP output only provides encoder position data, so the platform velocity must be computed from the empirical position data. The encoder output data file has four columns. The columns are defined as "Sample Number," "Time," "Commanded Position," and "Measured Position," respectively. This can be seen in Figure 32(a). Using MATLAB the encoder data is pulled from the text file created by the TCP and the measured data is pulled from the text file created by the RealTerm program. The sensors have been programed to output the data in a string of 10 data bytes, each one separated by a comma. The format of the sensor data string is defined as "Calibrated data/time(ms)," "Accelerometer X," "Accel. Y," "Accel. Z," "Magnetometer X," "Mag. Y," "Mag. Z," "Gyroscope X," "Gryo. Y," "Gryo. Z." Sample output of the IMU is shown in Figure 32(b).

(a)

```
| Sample    Time     Commanded Pos    Encoder 1 Pos
[    0     0.000            0.000           -1.969;
     1     0.009            0.000           -1.969;
     2     0.018            0.000           -1.969;
     3     0.027            0.000           -1.969;
     4     0.035            0.000           -0.469;
     5     0.044            0.031           -0.469;
```

(b)

```
#C=7661,27.58,24.62,245.13,31.94,-26.29,120.07,8.00,7.34,-14.27
#C=7682,25.65,25.60,246.12,32.51,-26.29,120.77,9.00,8.34,-13.27
#C=7702,27.58,24.62,243.15,33.65,-25.17,121.47,9.00,7.34,-14.27
#C=7723,27.58,24.62,243.15,33.27,-25.84,121.24,8.00,6.34,-13.27
#C=7743,27.58,24.62,245.13,32.32,-26.74,120.54,8.00,6.34,-14.27
```

Figure 32   Output text files: (a) TCP output file; (b) razor IMU output file

The MATLAB script calculates the velocity of the encoder using finite differencing. On average, for small sample times the finite differencing method can result in an in accurate solution. However, finite differencing adds noise to the data. The finite differencing result is shown as the blue dotted line in Figure 33.

To reduce the noise from finite differencing, MATLAB's curve fitting algorithm is used to approximate the velocity curve of the TCP platform. The approximated curve, in red, is shown superimposed on the blue finite difference velocity curve (see Figure 33(a)). Notice the peaks in the finite differenced velocity have been smoothed out. The smoothed velocity was integrated and plotted against the actual position of the encoder in order to test how well the smoothed curve approximated the real velocity (see Figure 33(b)).

51
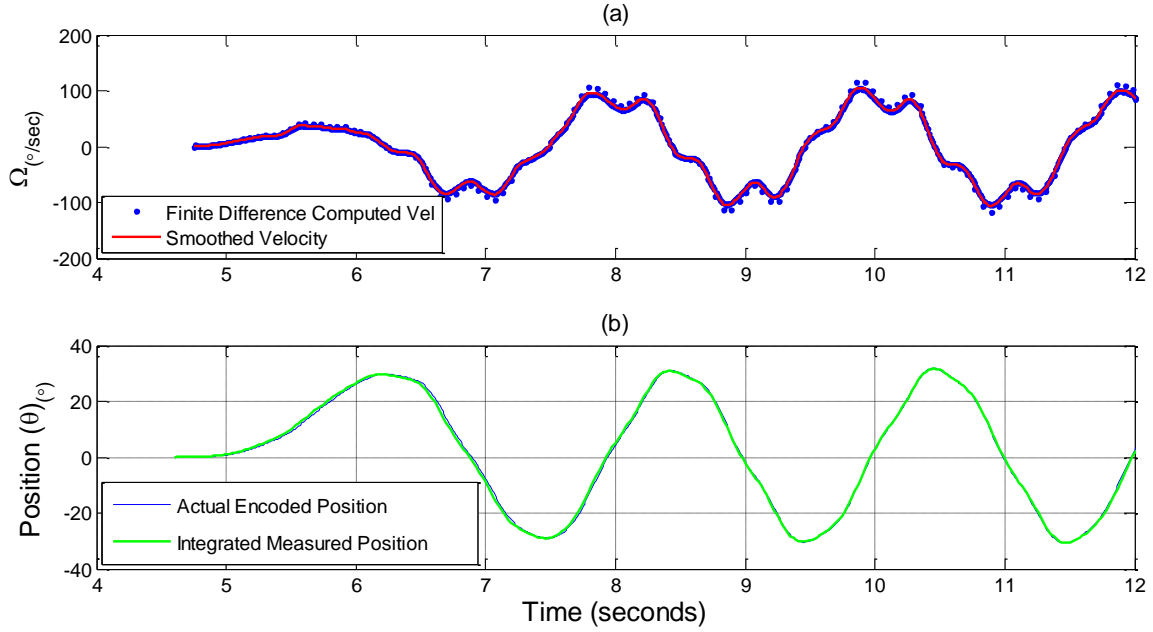
Figure 33   TCP single gyro motion experiment: (a) computed platform velocity;
(b) reconstructed platform position

The output of the MEMS gyros was integrated to compare the position estimate from the MEMS gyro to the actual encoded position from the TCP. The difference between the encoder position and the position estimation of the gyro is shown in Figure 34. The relatively small difference indicates that the MEMS gyro does a good job at approximating platform velocity and position.
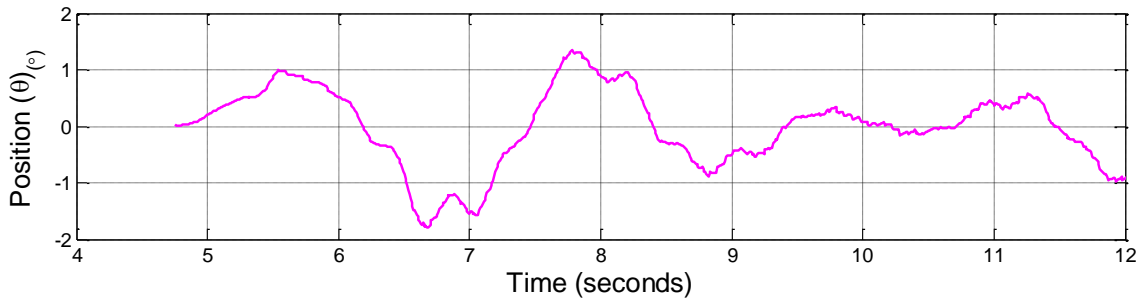
Figure 34   Error in estimated position for the single gyro TCP experiment

To validate the velocity output provided by the MEMS gyro, shown in green in Figure 35(a), it is compared to the curve fit solution, in blue. The difference is plotted in Figure 35(b). It appears that the timing of the two data sets may be causing a larger error then might be present if the two data sets were properly aligned. This issue can be corrected by running the test on a system with a common clock.
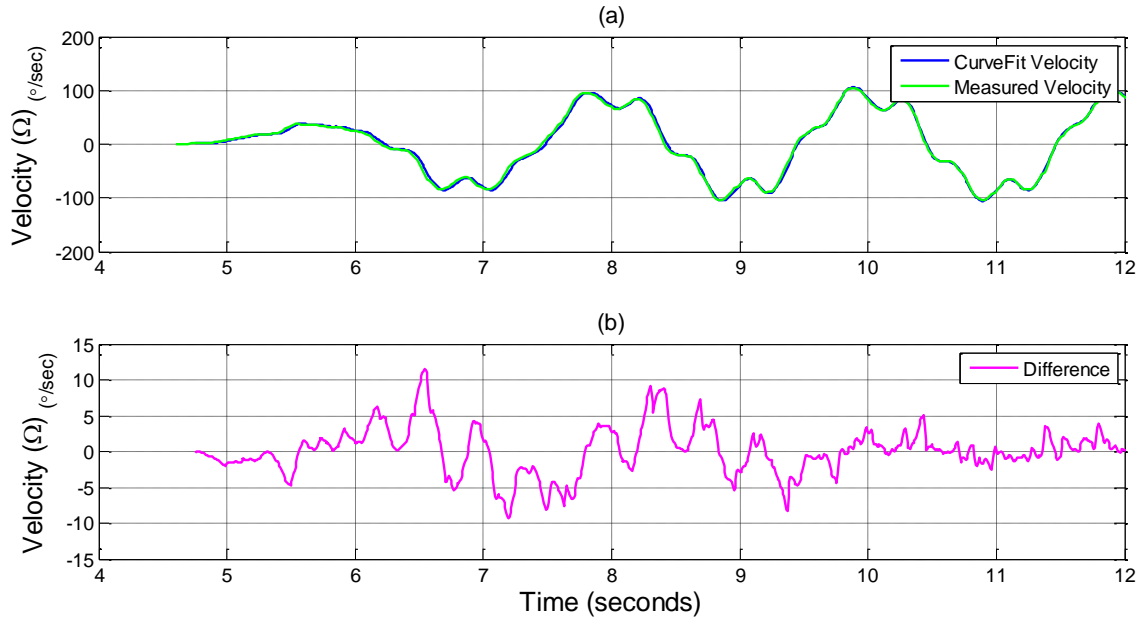
Figure 35   Single sensor test: (a) gyro velocity versus TCP platform velocity; (b) error in estimated velocity

This simple single gyro test verifies our ability to accurately acquire valid position and velocity data from the MEMS gyro and analyze that data using Excel and MATLAB. The test also shows that the MEMS gyro tracks the position and velocity of the TCP platform reasonably well. The position error is on the order of one to two degrees while the velocity error is about 10-15 °/s.

D.   MULTIPLE SENSOR VALIDATION

In an effort to move towards an array with more sensors, it was decided that an experiment using three sensors would be a useful step. This intermediate step helped

develop a method of managing the data from multiple sensors, and provides the first test case that validates the theoretical work done in Chapter II.

The set up was similar to the set up used for the single sensor case. However, in this case three sensors were powered and placed on the TCP as shown in Figure 36. The gyro configuration is laid out in the y-z coordinate plane with the individual measurement axes of each gyro labeled S^1, S^2, S^3.
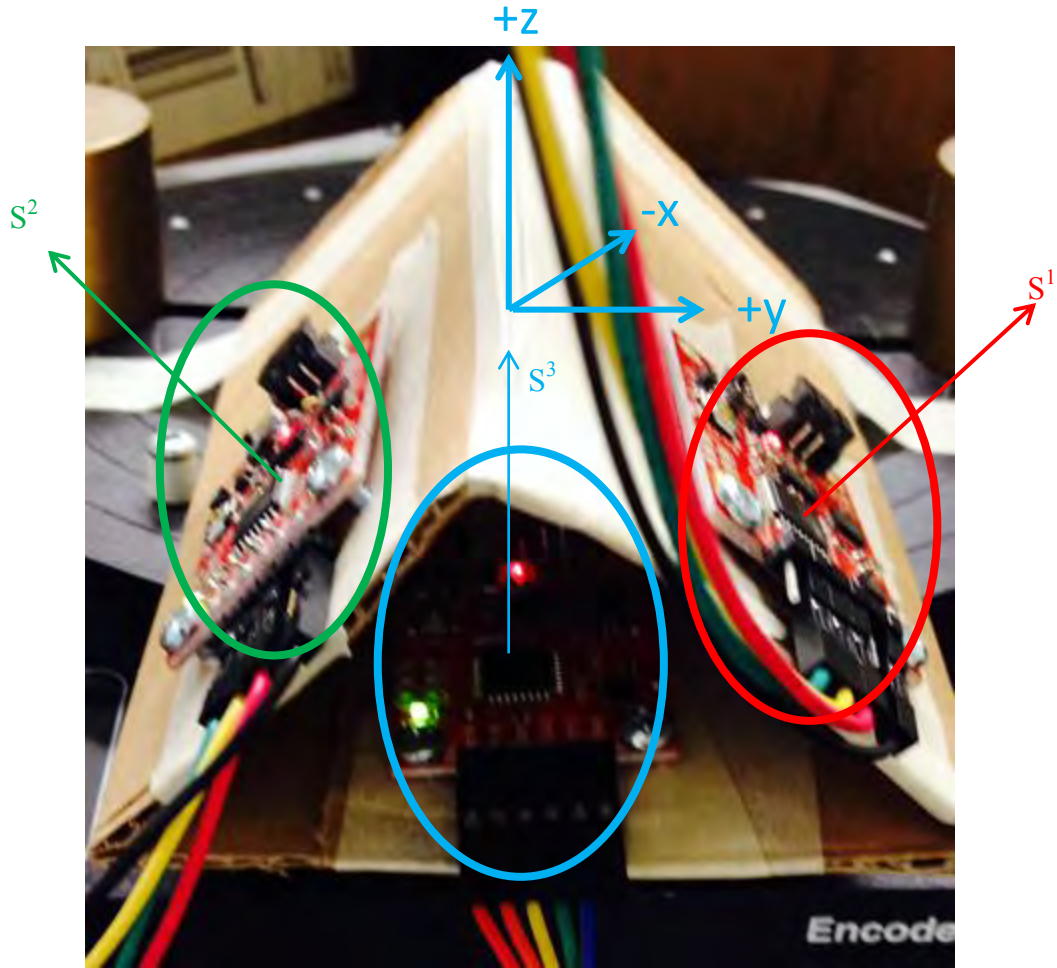
Figure 36   Three sensor configuration on the TCP

Sensors are named and organized as shown in Figure 37 with the +z-axis of sensor 1 pointed off in the +45 degree direction, the +z-axis of sensor 2 pointed off in

the -45 degree direction, and the +z-axis of sensor 3 pointed straight up along the +z-axis of the TCP platform. Figure 37 depicts the +z-axis alignments for each sensor in this array configuration.
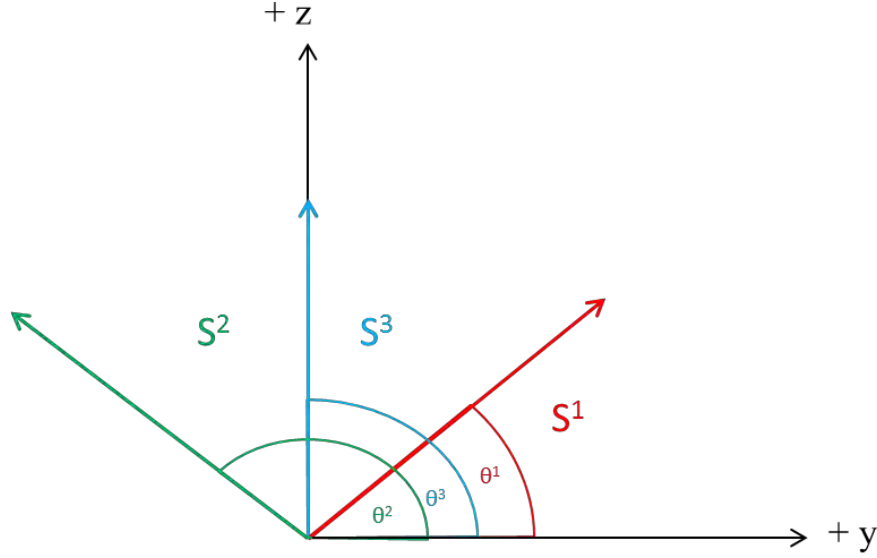
Figure 37   Nominal gyro alignments for the three sensor TCP test; θ^1=45°, θ^2=135°, θ^3=90°

The matrix that describes the gyros alignment is shown in (4.1).

$$
H = \begin{pmatrix} S^1\hat{i} & S^1\hat{j} & S^1\hat{k} \\ S^2\hat{i} & S^2\hat{j} & S^2\hat{k} \\ S^3\hat{i} & S^3\hat{j} & S^3\hat{k} \end{pmatrix} = \begin{pmatrix} 0\hat{i} & \cos(\theta^1)\hat{j} & \sin(\theta^1)\hat{k} \\ 0\hat{i} & \cos(\theta^2)\hat{j} & \sin(\theta^2)\hat{k} \\ 0\hat{i} & \cos(\theta^3)\hat{j} & \sin(\theta^3)\hat{k} \end{pmatrix} = \begin{pmatrix} 0 & \dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} \\ 0 & -\dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} \\ 0 & 0 & 1 \end{pmatrix} \qquad (4.1)
$$

Two tests were executed with the gyros oriented per (4.1). The first was a three minute still test. This was done to establish baseline bias for each of the three sensors. The individual sensor bias is estimated so that it can be subtracted out from the measurements obtained from later tests.

The measured rates from the z-axis of each gyro were combined using (4.2) to give the estimated rate about the platforms +z-axis.

55

$$\hat{\omega} = H^{-1}m \qquad\qquad (4.2)$$

The combined gyro measurements are compared to the individual z-axis rates for gyro number three. From Figure 38, it is apparent that the combined data is more precise than the data from gyro 3.
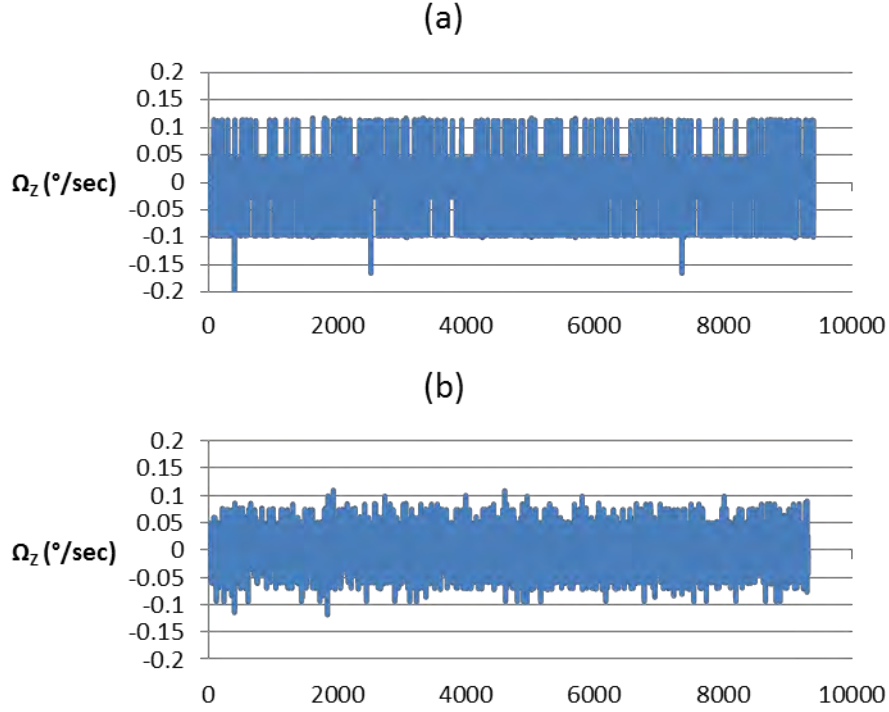
Figure 38   Measurement noise for: (a) gyro number three; (b) the combined array

Notice the reduced range and increased precision of the combined result when compared to the individual gyro. From these results it is clear that the addition of redundant gyros has helped to improve the precision of the array, at least qualitatively.

There is additional information that could be gathered from these results. The statistical data of the signal could be compared to the statistical data from analytical models. However, the signals from the MEMS gyros are transmitted in digital format; therefore the signals are discrete in nature and require statistical analysis that takes quantization effects into consideration. From the sensitivity scale factor found in [6], the gyro has a resolution of 0.069 (°/s/bit). Thus the, noise present in the output of the

individual rate gyro, see Figure 38(a), appears to be quantized (i.e., one or two bits flipping back and forth). Because the noise floor is close to the quantization level, it becomes necessary to look into how quantization impacts the noise statistics.

1.   Quantization of Noise

The initial results presented in this chapter showed that there is an apparent improvement to the precision of an array when the number of sensors is increased. Another way to measure improvement of the system would be to compare the sensors noise with that of the combined array. The data analyzed was clearly affected by digital noise. This was evident by the fact that the sensors remained still but their digital outputs indicated that they were moving sporadically around their actual zero rates.

In order to characterize the random noise of the gyro, it is necessary to perform statistical analysis on its output. However, in this case the signals are digital. To run the same statistical analysis that was performed in Chapter II on the digital data would be incorrect. This is because that analysis does not take the quantized nature of the output signal into account. Therefore, the results may not truly reflect the statistics of the real data. The theory developed in [32] details the assumptions and methods for statistically analyzing quantized signals. In short, it is possible to compute the statistics of the quantized signals by applying the corrections (4.3), to the standard calculations.

$$Mean(\mu): E\{x\} = E\{x'\} - (0)$$

$$STD(\sigma^2): E\{x^2\} = E\{(x')^2\} - (\frac{1}{12}q^2)$$

$$Skew(\gamma): E\{x^3\} = E\{(x')^3\} - (\frac{1}{4}E\{x'\}q^2)$$

$$Kurtosis(\kappa): E\{x^4\} = E\{(x')^4\} - ((\frac{1}{2}E\{x'\}q^2) - \frac{7}{240}q^4)$$

(4.3)

where q is the quantization number equal to the resolution of the gyro, 0.06965 (°/sec/bit).

2.   Statistical Noise Analysis

Equations in (4.3) provide the continuous equivalent statistics for the quantized gyro outputs. The digital signals are only able to represent angular rates within a

precision of plus or minus half of the gyros resolution. As discussed previously, the resolution for the gyro is 0.069 °/sec/bit. Therefore the sensor cannot detect angular rates between 0 and 0.069 °/sec [6]. This creates a lack of data points between the resolution gaps of the gyro. To estimate the data in between the resolution gaps, the continuous equivalent statistical properties are calculated. The resulting statistical analysis can be used to evaluate and benchmark the capability of each sensor and then compare it to the combined results of the array.

The data displayed in Figure 39 is a result of the three sensor array held stationary for 30 minutes. A histogram for each gyro is displayed along with a normal distribution curve reconstructed as per (4.3).
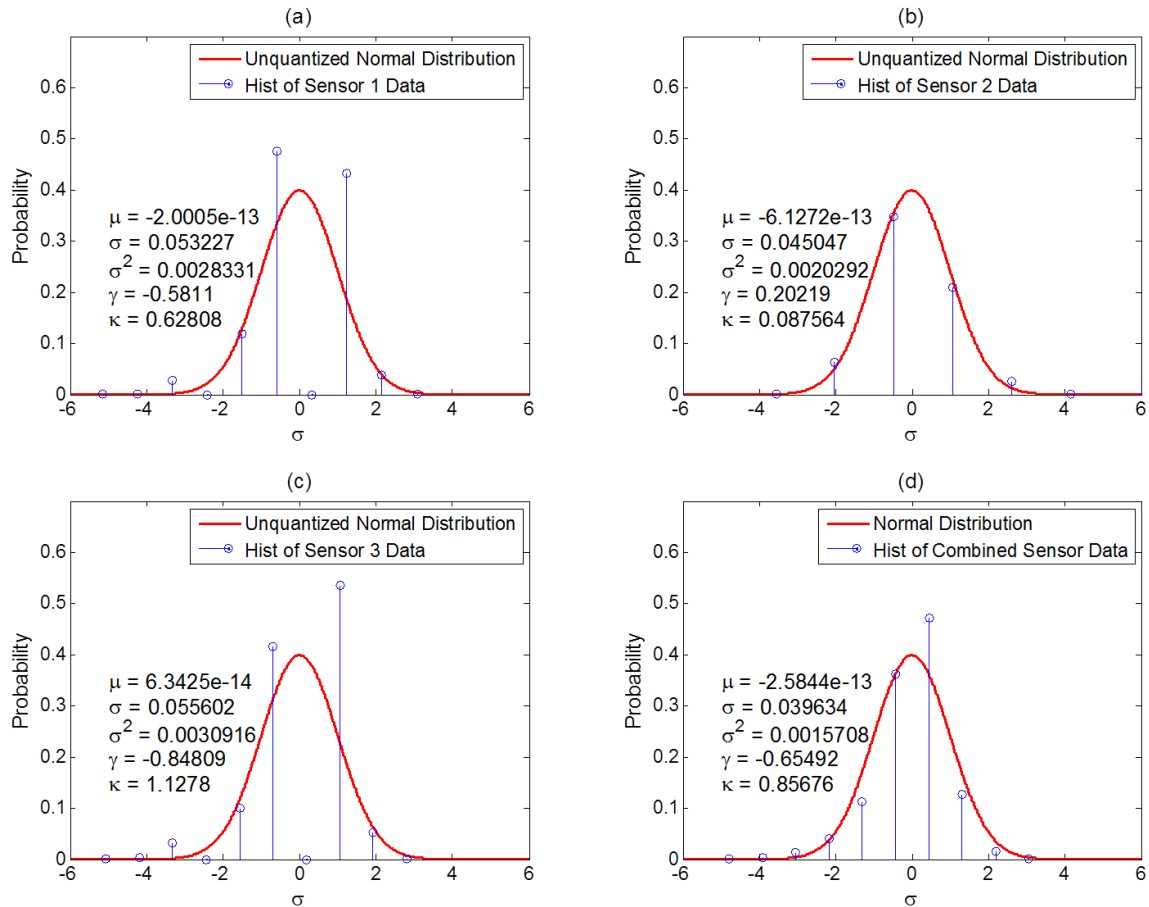

Figure 39   Statistical results: (a) gyro #1; (b) gyro #2; (c) gyro #3; (d) combined array

The quantized nature of the individual sensor histograms only shows data observations in the digital output bins. All the sensors have similar signal characteristics that factor in the combined sensor result shown Figure 39(d). The precision of the combined array is clearly an improvement over the precision denoted by any of the individual gyros. The best 1-σ deviation for any of the individual gyros was 0.045 (°/sec) while the combined array improved this statistic to 0.039 (°/sec). That is a 13.3 percent improvement with the addition of just two sensors.

Recall from Chapter II that the variance for the combined array is given by (2.8), the development of which is repeated here for clarity. The covariance matrix R comprises of the individual variances for each gyro, and the H matrix is a function of the geometry of the array. For this test the R, H, Var matrices are shown below.

$$R = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} = \begin{pmatrix} 0.0028 & 0 & 0 \\ 0 & 0.0020 & 0 \\ 0 & 0 & 0.0031 \end{pmatrix} \tag{4.4}$$

$$H = \begin{pmatrix} S^1\hat{i} & S^1\hat{j} & S^1\hat{k} \\ S^2\hat{i} & S^2\hat{j} & S^2\hat{k} \\ S^3\hat{i} & S^3\hat{j} & S^3\hat{k} \end{pmatrix} = \begin{pmatrix} 0\hat{i} & \cos(\theta^1)\hat{j} & \sin(\theta^1)\hat{k} \\ 0\hat{i} & \cos(\theta^2)\hat{j} & \sin(\theta^2)\hat{k} \\ 0\hat{i} & \cos(\theta^3)\hat{j} & \sin(\theta^3)\hat{k} \end{pmatrix} = \begin{pmatrix} 0 & \dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} \\ 0 & -\dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} \\ 0 & 0 & 1 \end{pmatrix} \tag{4.5}$$

$$Var(\hat{\omega}) = (H^T R^{-1} H)^{-1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.0024 & 0 \\ 0 & 0 & 0.0014 \end{pmatrix} \tag{4.6}$$

From Figure 39(d), the experimental measured z-axis variance is $\sigma^2 = 0.0015$ (°/sec). The theoretical z-axis variance given in (4.6) is very similar to the experimental variance computed from the experimental data. Since the two results correlate, the theoretical variance seems to be a very good approximation for the actual variance, and can be used to estimate performance improvements in multi-dimensional gyro arrays.

To further explore the properties of the multi gyro array, the same TCP 30º motion wave experiment was conducted again with the multi gyro array. The resultant velocity curve from the combined three gyro measurement is compared to the actual encoded velocity derived from the TCP encoder. The data is formatted in Excel and imported into MATLAB for the same curve fitting and difference analysis that was done for the single sensor validation. Figure 40 shows the results from the three gyro TCP motion experiment..

Figure 40   Three gyro TCP motion experiment: (a) computed platform velocity;
(b) reconstructed platform position

The previous statistical analysis results created an expectation that this experiment would show decreased error in the estimated position. Surprisingly, the error increased, as illustrated by comparing Figure 34 and Figure 35 with Figure 41 and Figure 42.
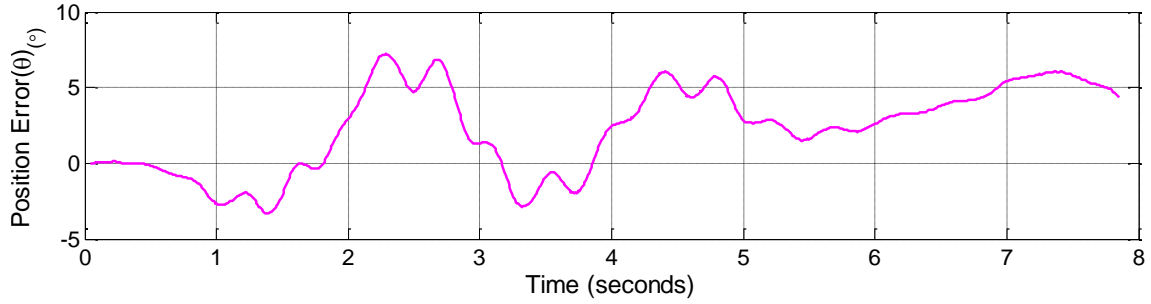
Figure 41   Error in estimated position for the three gyro TCP experiment

The error in the signals shown above is exaggerated more than it was in the single gyro test because the clocks on each gyro and the clocks of the TCP encoder are not accurately aligned. The timing error is a result of the RealTerm software's inability to synch the timing of the data, against the clock of the TCP system. This error reduces our ability to make an accurate assessment of the benefit of the redundant measurements.
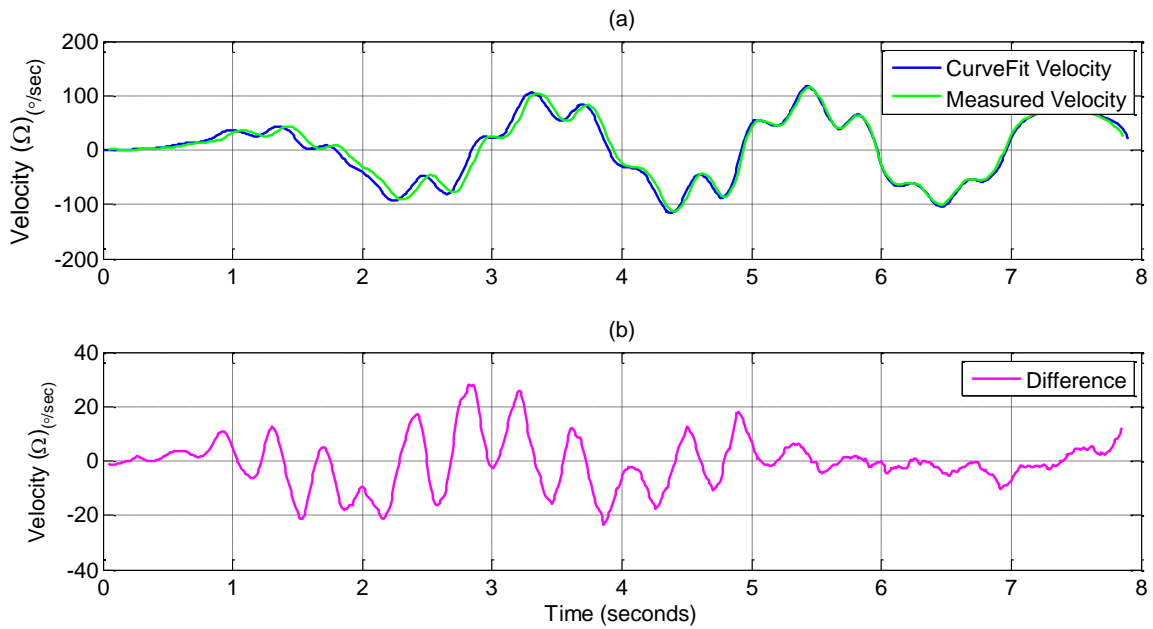
Figure 42   Three sensor test: (a) gyro velocity versus TCP platform velocity; (b) error in estimated velocity

Handling the synchronization and timing of real hardware is a very challenging aspect associated with multiple sensors. A more robust sampling system is required in

order to realize the increased precision of the combined array. This issue will be corrected by running the test again using the cRIO system. The cRIO will provide near real time sampling with a common clock for all the MEMS gyros.

3.   cRIO Validation

Using the cRIO to take samples of every sensor at the same time should resolve the time synch issue identified in the previous section. The same 30° sine wave experiment is conducted using the cRIO as the sampling device. The cRIO uses the code developed in LabBIEW to synch and record the data streams from all three gyros. The data was processed the same way using Excel and MATLAB. The position and velocity results are shown in Figure 43 and Figure 45, respectively.
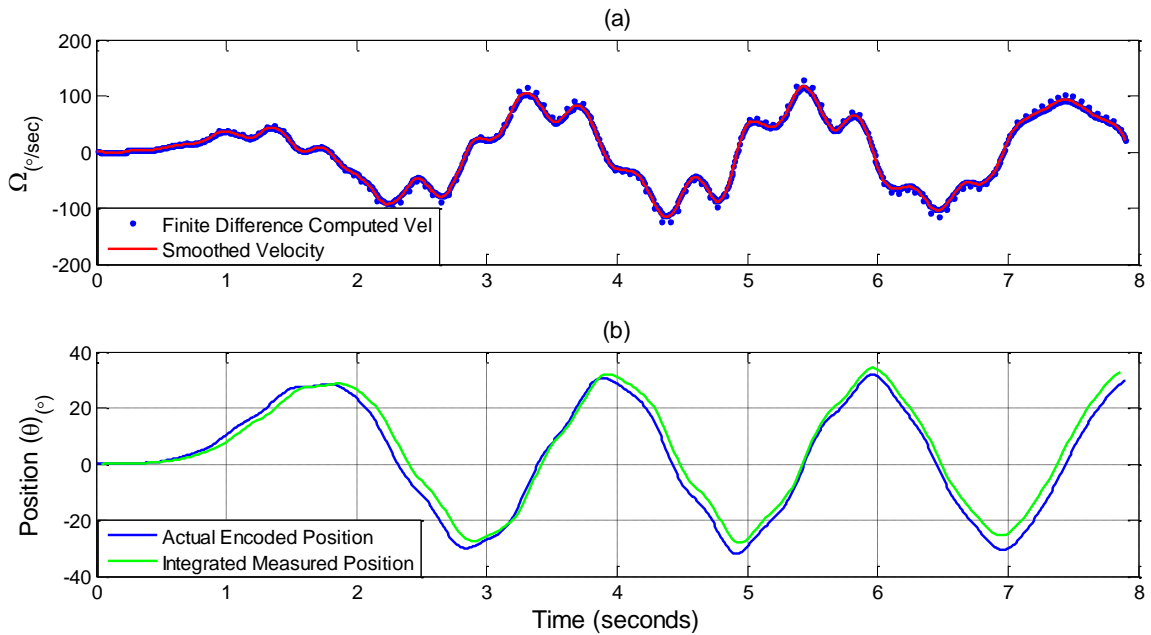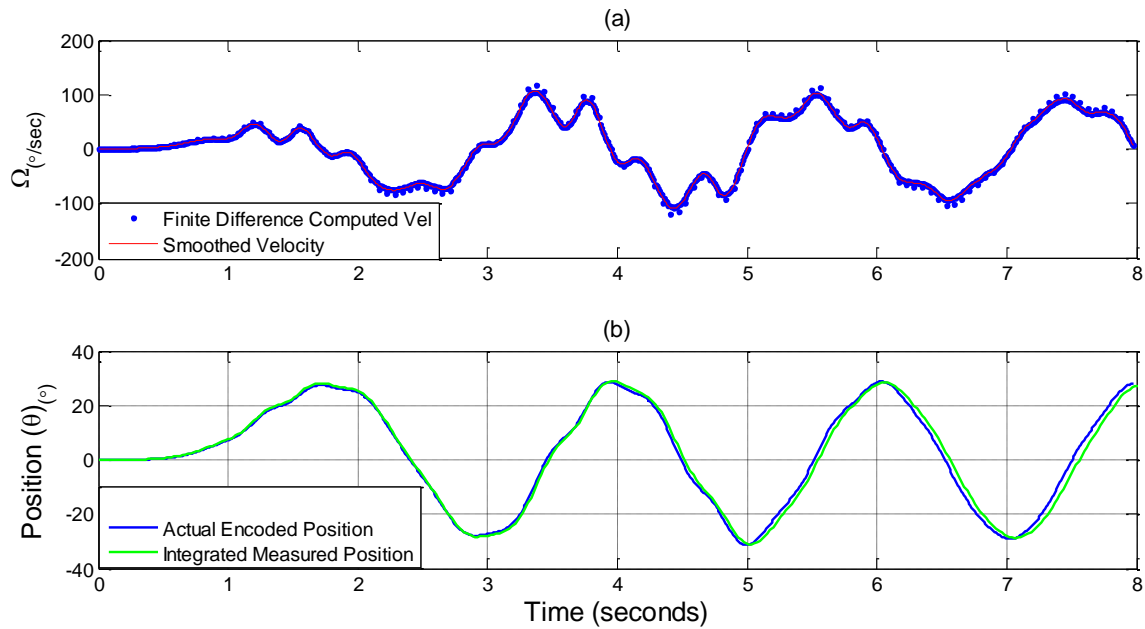

Figure 43   cRIO three gyro TCP motion experiment: (a) computed platform velocity; (b) reconstructed platform position

After the cRIO was used the zero crossings of the data line up better. This is an indication that both the data streams and the timing of the sampling are synched. The results indicate an improvement over Figure 41 and Figure 42, but as evident in Figure 44, still show a larger error than in the single sensor test.
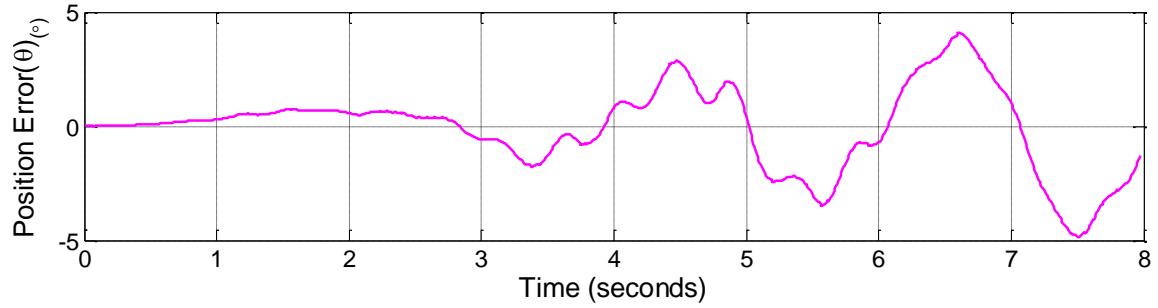
Figure 44   Error in estimated position for the cRIO three gyro experiment
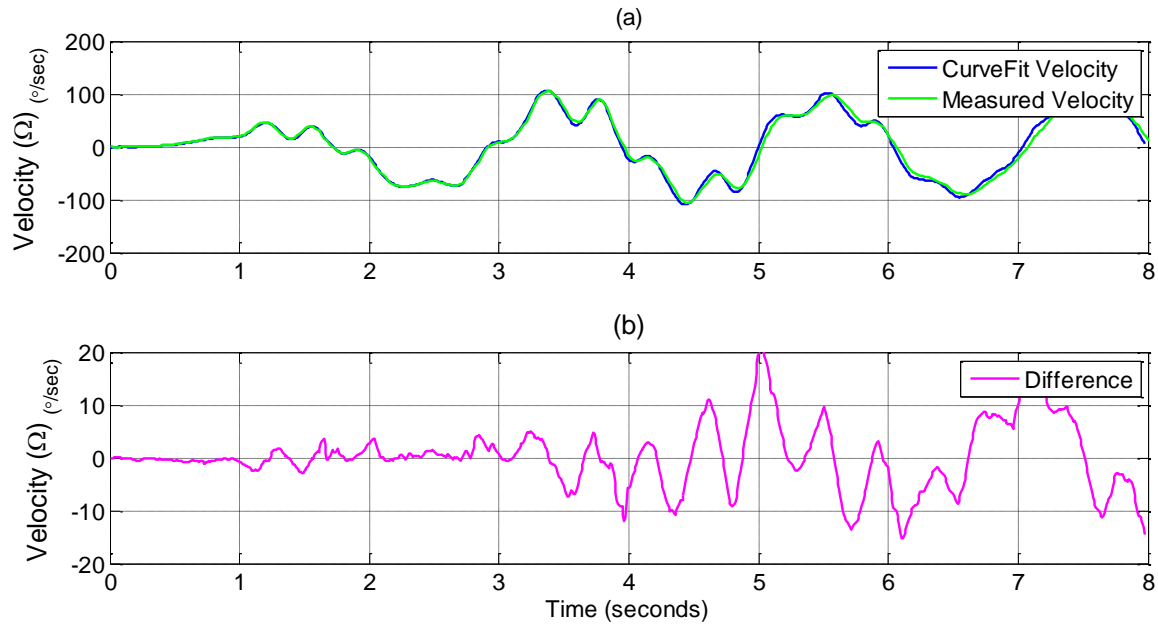
Figure 45   cRIO three sensor test: (a) gyro velocity versus TCP platform
velocity; (b) error in estimated velocity

The TCP output was not measured using the cRIO. Therefore, the increasing error is most likely sourced from the inaccurate clock on board the TCP encoder. However, bias estimation may also be playing a role. Proper estimation of the sensor bias is crucial for practical implementation of a multi-sensor array. Estimating the bias is beyond the scope of this thesis, but improving the noise statistics using the ideas presented here will have a key role to play in this process.

E. SUMMARY

This chapter presented the initial testing and calibration of the Sparkfun razor IMU. Experiments using a single gyro were performed in order to develop a method for gathering and analyzing data on three gyros. The majority of the work was focused on interfacing the gyros with the cRIO. This was done to reduce the errors associated with timing. Once the data was obtained it was also important to understand how to handle and quantify the performance of a digital MEMS gyro. A 13 percent reduction in noise was obtained through the addition of two gyros. The next chapter will use the test bed developed in Chapter III to expand the experiment to 12 gyros, with a specific focus on gyro noise output.

V. EXPERIMENTS WITH THE LARGE SENSOR ARRAY

A. INTRODUCTION

This chapter presents the methods and techniques used to obtain results from the 12 MEMS gyro experiment. The details of developing the sensor matrix as well as the approach for selecting the ideal sensor combination will be explained. The experimental results will be compared to those obtained from the previous experiments as well as the theoretical expectations.

B. DEVELOPING THE CONFIGURATION (*H*) MATRIX

As explained in Chapter II, the configuration matrix, *H*, is used to describe the geometrical arrangement of all the sensors. Creating the *H* matrix for the 12 gyro sensor array requires some effort. First it was necessary to model the icosahedron geometry. To create a three dimensional icosahedron it was necessary to accurately locate all 12 vertices in three dimensional space. From [33], the coordinates of the 12 vertices for a unit-icosahedron are defined as shown in Table 1, where

$$\varphi = \frac{1 + \sqrt{5}}{2} \qquad (5.1)$$

Table 1   Coordinates of the 12 vertices for a unit-icosahedron

Vertices	Coordinate [x, y, z]
1	[0, 1, φ]
2	[0, -1, φ]
3	[0, 1, -φ]
4	[0, -1, -φ]
5	[1, φ, 0]
6	[-1, φ, 0]
7	[1, -φ, 0]
8	[-1, -φ, 0]
9	[φ, 0, 1]
10	[-φ, 0, 1]
11	[φ, 0, -1]
12	[-φ, 0, -1]

Once all the vertices were located it was necessary to connect them with edges to form the faces. Since each face is an equilateral triangle, any single face can be identified by its three vertices. Each face was identified and labeled 1-20 as shown in Figure 46. One MEMS IMU can be mounted to each face of the icosahedron. However, only 12 of these faces will be used in this experiment. Moreover, the orientation of each gyro with respect to a common reference frame must be determined.
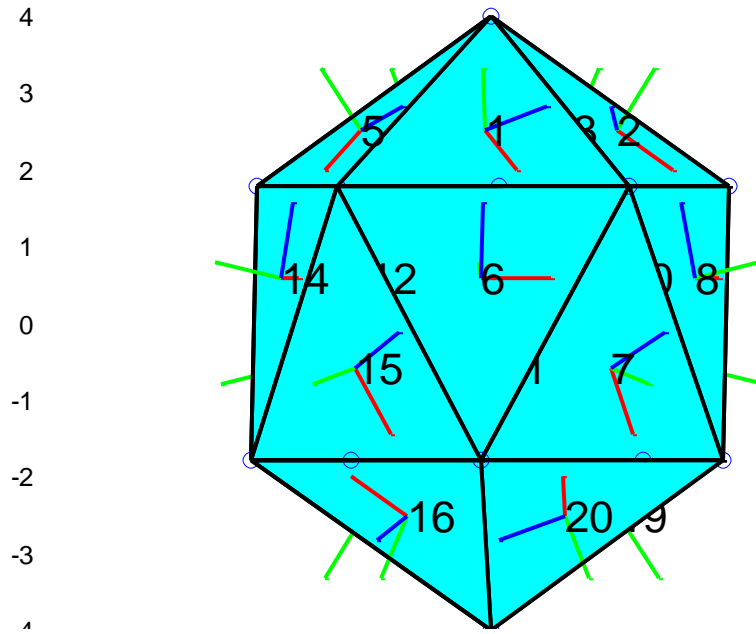
Figure 46   3-D model of the icosahedron test bed

To model the orientation of each gyro it was necessary to develop a common mounting scheme for the actual gyros. The gyros were mounted in a way that aligns the x-axis of the gyro with a specific edge of the face it was to be mounted on, with the +z-axis normal to the face. This mounting scheme is shown in Figure 46, the +x and +z axes are shown as red and green lines respectively. The orthogonal coordinate system framed by the red, blue, and green axis on each face represents the local coordinate frame of each individual gyro. In order to accommodate rotation of the sensor panels, the +x-axis should be rotated to be aligned with the appropriate edge of each face.

The local reference frame for each gyro was modeled in MATLAB by first creating the x-axis. This was done be defining an edge of a face as the local x-axis. Then another edge of a face is defined and used to take the cross product with the x-axis. The result of the cross product is a vector normal to the face as shown in Figure 47(a).
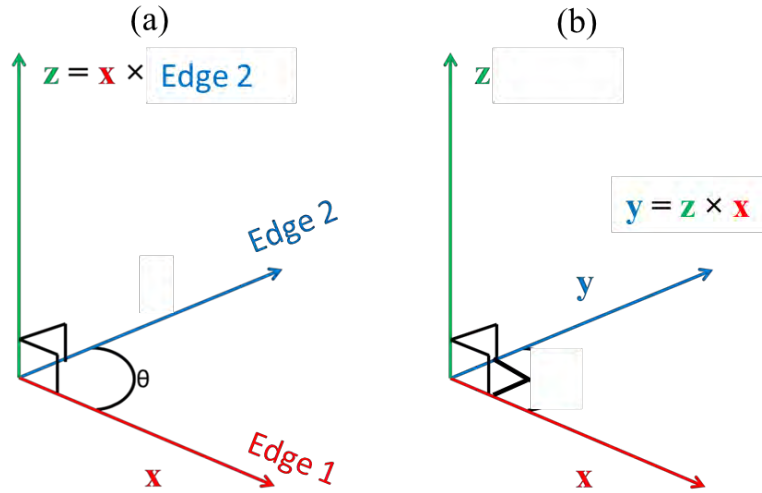
Figure 47   Cross product example: (a) identifying +z-axis; (b) identifying +y-axis

This green vector in Figure 47(a) represents the +z-axis of the gyro. The cross product is used again, in Figure 47(b), with the x and z axes to define the direction of the +y-axis. Each vector was normalized and plotted in the center of each face. This process is repeated for each of the 20 faces of the test bed, to create the local gyro coordinate systems displayed in Figure 46. The pseudo-code in Figure 48 describes the process of computing each frame as it was performed using MATLAB script.

```
This code is run 20 times, once for each face of the icosahedron

Creating the X-axis
        Edge of a face is defined by two vertices
        X-axis = edge of face (normalized)

Creating the Z-axis (normal)
        Define other edge of face with other vertices
        Other edge=edge of other face
        Calculate the cross product between X-axis & other edge
        Z-axis = cross product of (X-axis, other edge) (normalized)

Creating the Y-axis axis
        Take cross product between X-axis and Z-axis
        Y-axis= cross product of (X-axis, Z-axis) (normalized)
```

Figure 48   Pseudo-code for computing gyro reference frames

The orientation of each of these local gyro frames with respect to a common frame is used to construct the H matrix for each gyro. For this experiment the common reference frame was chosen to be the local frame of gyro #1. Gyro #1 was placed on face number 1 as shown in Figure 49.
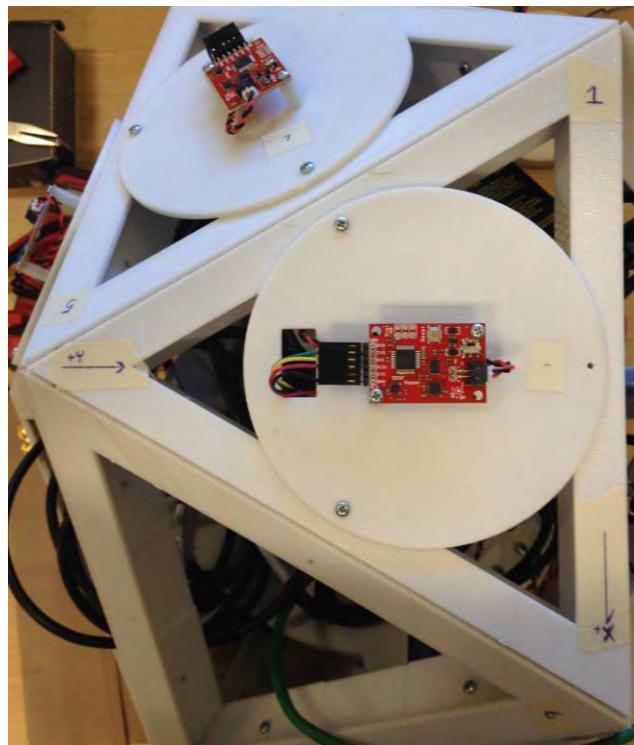
Figure 49   Sensor number one on face number one

Gyro 1 is the only sensor that corresponds to its face number. Figure 46 shows the modeled test bed with the assigned face numbers. Figure 49 shows the actual test bed with individual gyros on each panel. Gyro numbers 1 through 12 are each placed on one of the possible 20 faces the test bed offers. They are aligned as shown in Figure 46. To keep the gyro placement and configuration consistent with the *H* matrix it was necessary to create the mapping table shown in Table 2. Table 2 details the identification of each sensor with respect to the panel it is mounted on. This mapping table is necessary to match raw sensor data to the correct columns of the *H* matrix, which is built based on the geometry of the icosahedron.

Table 2   Gyro mapping table

Gyro #	Face #
1	1
2	14
3	13
4	12
5	4
6	3
7	5
8	7
9	20
10	16
11	9
12	19

Using the local *H* matrix, each gyro axis is then projected onto the common reference frame. The projection results in an *H* matrix for each face $H_1 \rightarrow H_{20}$, referenced to face 1. The *H* matrices, of the individual faces that correspond to a gyro, are combined and stored into the combined *H* matrix that will be used to process data from the combined sensor array:

$$H = \left[H_1; H_2; H_3; H_4; H_5; H_6; H_7; H_8; H_9; H_{10}; H_{11}; H_{12} \right] \tag{5.2}$$

Previous experiments showed an improvement in gyro performance when the number of redundant sensors was increased from 1 to 3. Additional improvement is

expected as the number of gyros increases. However, it was also noticed that if additional gyros are added in the non-ideal alignments then a decrease in precision would result. Therefore, a method of identifying the best H matrix for a given number of gyros must be found. The approach for doing this in this experiment is described in the next section.

C.   FINDING THE IDEAL COMBINATION OF SENSORS

In order to identify the ideal configuration for a given number of gyros, a MATLAB script was generated to create and check every possible gyro configuration allowed by the icosahedron test bed. Given the number of gyros desired, one to 12, the configuration that creates the best performance is found. The best performance is found by searching for the sensor combination that minimizes the trace of the covariance matrix. The result of the process is the best configuration for any number of gyros used. Another script was developed to process the combined output using the raw gyro data. The pseudo-code describing the exhaustive search algorithm that was used is shown in Figure 50.

```
This code is to be ran a total of 12 times, once for each combination of
gyros shown below.

   1st run using 1 Gyro
   2nd run using 2 Gyros
   3rd run using 3 Gyros
   .
   .
   .
   12th run using 4 Gyros

Initialize gyro sigma to 0.04 (this is the average individual gyro sigma)
Input the number of gyros to N= (1-12)
Compute 12 choose N to determine the total number of combinations to check

For each combination found above
     Create an H matrix
     Calculate the Variance
     Compute the Trace of the Variance matrix
     Compare Trace of current configuration to the minimum Trace
     Save the resulting configuration as the minimum Trace

Output the configuration in terms of an expected accuracy for the array
(analytical sigma) and the panel numbers that represent the H matrix that
resulted in the minimum Trace.
```

Figure 50   Pseudo-code for finding ideal sensor combinations

Table 3 displays the results of the ideal sensor configuration identified using the approach outlined in Figure 50. The 1-σ values displayed are the average between all three axes of the combined array. The percent improvement, *IM* is calculated by

$$IM = \frac{\sigma^N - \sigma^1}{\left|\sigma^1\right|} \times 100 \qquad (5.3)$$

where σ^N is the 1-σ value of N gyros in an array and σ^1 is the 1-σ value of 1 gyro.

Table 3   Expected 1-σ values for the multi-gyro array

# of Gyros	1-σ (°/s)	Panel #s	Expected % Improvement
1	0.0400	1	n/a
2	0.0283	5,9	29.25
3	0.0231	3,5,16	42.25
4	0.0200	3,5,12,13	50
5	0.0179	3,4,5,12,16	55.25
6	0.0163	1,3,12,13,14,16	59.25
7	0.0151	3,4,5,9,12,13,16	62.25
8	0.0141	1,3,4,5,7,9,13,16	64.75
9	0.0133	3,4,5,7,9,12,13,14,16	66.75
10	0.0126	1,3,4,5,7,9,12,13,14,16	68.5
11	0.0121	1,3,4,5,7,9,12,13,14,16,19	69.75
12	0.0115	1,3,4,5,7,9,12,13,14,16,19,20	71.25

The results shown in Table 3 illustrate the expected improvement in precision and describe the ideal configuration with face numbers that can be obtained on the test bed. The next section will use the mapping in Table 2 with the panel numbers given in Table 3 to verify that the experimental results are in line with the expected results.

D.   SENSOR ARRAY EXPERIMENTS

There were 12 individual gyros at the start of the experiment. However, during the statistical analysis, it was noticed that the statistics for sensors 11 and 12 were not consistent with the other devices. After closer inspection, it was clear that the data from the gyros 11 and 12 did not correctly map into the organized format they were programed to. This mapping error resulted in some magnetometer data getting confused with the gyro data. The large magnetometer numbers in series with the gyro data created outliers in the statistics data. This error can be detected and corrected in the future with a data checking algorithm. The code developed for this thesis does not incorporate an error checking algorithm. Therefore, data from gyros 11 and 12 will be ignored for the remainder of the present analysis.

To accurately measure the improvement in performance achieved by the combined array, it was necessary to implement a filtering technique to estimate the gyro

bias. The filter estimated the bias, on each axis of every gyro, as it changed throughout the 1 hour experiment. The bias was then subtracted out from each axis. The filter used was a moving average filter that operates by averaging a number of points from the input signal to produce each point in the output signal.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \qquad (5.4)$$

Equation (5.4) is the moving average filter. In this equation, *x[]* is the input rate signal from a given gyro channel, *y[]* is the filtered output, and *M* is the number of points used in the moving average [34]. Equation (5.4) was implemented using the MATLAB "filter" command. An example of the estimated bias for one of the gyros is shown in Figure 51.
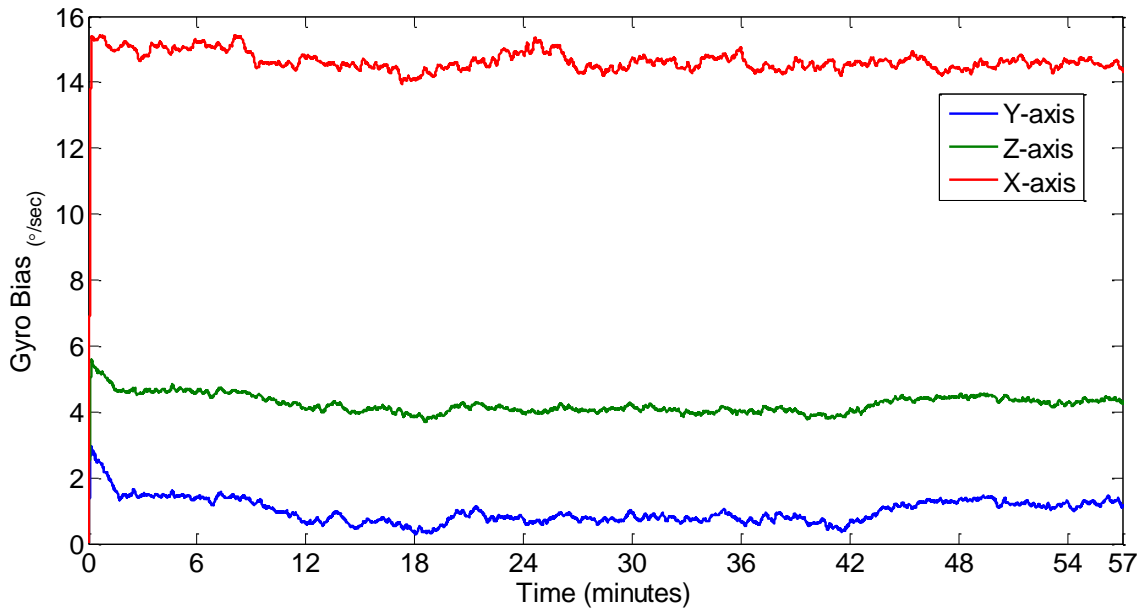

Figure 51   Estimated gyro bias

The de-biased data was combined as dictated by the ideal gyro configuration codes described in Table 3. The statistics of the combined measurements are shown in Table 4.

Table 4   Experimental 1-σ values for the multi-gyro array

# of Gyros	1-σ (°/s)	Gyro #s	Experimental % Improvement
1	0.0449	1	n/a
2	0.0308	7,11	23.0
3	0.0253	6,7,10	36.8
4	0.0218	3,4,6,7	45.5
5	0.0196	4,5,6,7,10	51.0
6	0.0183	1,2,3,4,6,10	54.3
7	0.0163	3,4,5,6,7,10,11	59.3
8	0.0155	1,6,5,7,8,11,3,10	61.3
9	0.0146	2,3,4,5,6,7,8,9,10	63.5
10	0.0141	1,2,3,4,5,6,7,8,9,10	64.8

These experimental results are nearly identical to the expected results as shown in Figure 52, which compares the predicted and experimental results. The experimental results demonstrate that the precision of an array can be increased by adding redundant sensors in an ideal orientation. To further illustrate the point, Figure 53 shows the raw data of the gyro 1 compared to the processed output of the 10 gyro array. The data is displays a level of improvement consistent across all axes.
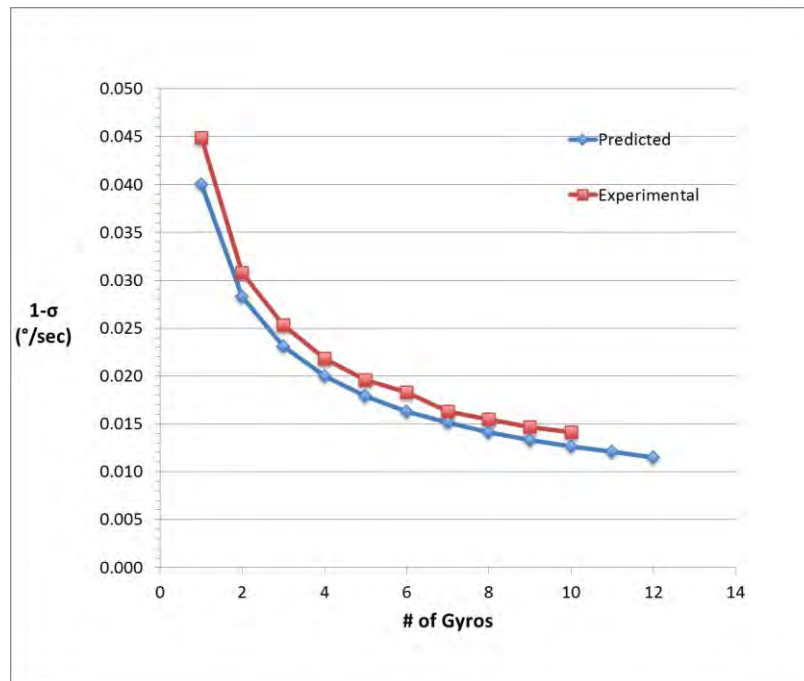
Figure 52   Analytical vs. experimental results
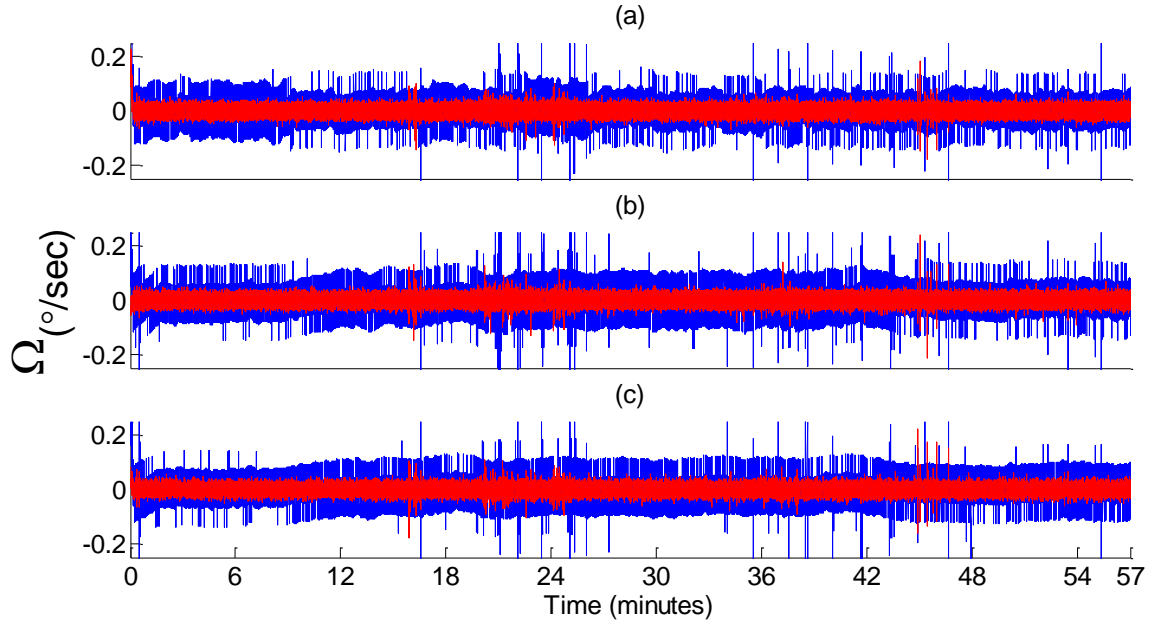
Figure 53   Noise output of a single gyro (blue) vs. the noise output of a 10 gyro
array (red): (a) x-axis; (b) y-axis; (c) z-axis

E.   SUMMARY OF RESULTS

An exhaustive search method of optimization was implemented to find the ideal combination of multiple gyros. This approach allowed for the improvement in precision as the number of gyros was increased. The experiments validate the theory explained in Chapter II and showed the 1-σ noise values could be reduced by 65 percent when 10 gyros are used in an array. The results also show an exponential decay that reduces the overall improvement as the number of sensors continues to increase.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

A. RESEARCH SUMMARY

This thesis explored the possibility of improving the performance of an array of MEMS gyros as compared to a single gyro. It was conjectured that by adding redundant gyros in an ideal configuration it would be possible to increase the precision of the array. The results of Chapter II demonstrated that in theory, performance of an array could be improved through the use of redundant sensors. The analysis showed that it is possible to improve array performance. This is only true if the redundant sensors are added in an ideal configuration.

A specialized embedded system was developed for data acquisition from multiple IMUs to support the experimental work of this thesis. The test bed can also be adapted to suit a number of further research needs. Experiments using the developed test bed were conducted to validate the theoretical results of Chapter II. Multiple issues, such as noise quantization and data synchronization must be considered when working with real hardware. If these issues are not taken into consideration, their effects can negate any improvements in sensor array performance.

Both analysis and experiments showed that that the precision of an array of MEMS gyros can be improved through the combination of redundant sensors. In particular, experiments with a multi-sensor test bed showed a 65 percent increase in precision when 10 gyros are properly configured and used as a redundant sensor array.

B. FUTURE WORK

In addition to the gyro, each Sparkfun razor IMU has on board an accelerometer and magnetometer. The data from those two sensors was not utilized in this thesis. Continued research into how the data from these sensors can be used to increase performance should be done. In particular, the outputs of the sensors should be used to estimate the gyro bias, which cannot be simply averaged out if the system is moving.

This thesis used an exhaustive search over a predefined structural geometry to determine the configurations that resulted in the greatest improvement to sensor array performance. More research should be done that focuses on developing an optimum geometry. Without the limitation of a predetermined sensor geometry, there is potential to drive the test bed construction based on the ideal geometric configuration described by an optimal sensor configuration matrix.

Allan Variance analysis should also be performed for the multi-sensor array. The ARW should improve as precision improves. Therefore, it can be expected that the ARW will also improve for the 10 gyro array. Continued work on characterizing the ARW on this system will present the realized performance improvement as it relates to a common industry standard.

To further explore the performance of a multi-sensor system, future work should focus on using a high-fidelity, moveable platform for experiments with a moving array. The high-fidelity single axis air bearing shown in Figure 54 is an example of such a device. The high quality linear motor and optical encoders will deliver outstanding performance and provide a more accurate motion base for future experiments. This equipment is available in the control and optimization laboratories at the Naval Postgraduate School.

Figure 54   Motorized linear air bearing available at Naval Postgraduate School

LIST OF REFERENCES

[1]   Quartz MEMS inertial sensing products and systems. (n.d.). Unmanned Systems Technology. [Online]. Available: http://www.unmannedsystemstechnology.com/company/systron-donner-inertial/ Accessed Mar. 10, 2014.

[2]   J. W. Judy, "Microelectromechanical systems (MEMS): fabrication, design and applications," *Smart Materials and Structures*, vol. 10, no. 6, pp. 1115–1134, Nov. 2001.

[3]   C. C. Naranjo, "Analysis and modeling of MEMS based inertial sensors," Kungliga Tekniska Kgskolan, Stockholm, School of Electrical Engineering (2008).

[4]   J. Geen. (2003, Nov. 8). New iMEMS angular rate sensing gyroscope. [Online]. Available: http://www.analog.com/library/analogdialogue/archives/37-03/gyro.html

[5]   U. S. Dynamics Corporation, "Rate integrating rotary gyroscope," U. S. Dynamics Corporation., Amityville, NY, Tech. Rep. 446225, Aug. 2006.

[6]   InvenSense., "ITG-3200 product specification, MEMS gyro IC," InvenSense, San Jose, CA, Tech. Rep. PS-ITG-3200A-00-01, Jul. 2010.

[7]   Silicon sensing state of the art MEMS gyroscope. (n.d.). Silicon Sensing MEMS Gyroscpes. [Online]. Available: http://www.siliconsensing.com/products/mems-gyroscopes/. Accessed Mar. 10, 2014.

[8]   KVH, "DSP-3000 FOG high-performance, single axis fiber optic gyro," KVH Inc., Middletown, RI. Tech. Rep. DSP-3000, Jul. 2009.

[9]   E. Linacre and B. Geerts. (n.d.). The Coriolis acceleration. [Online]. Available: http://www-das.uwyo.edu/~geerts/cwx/notes/chap11/coriolis_acc.html

[10]   P. Greiff et al., "Silicon monolithic micromechanical gyroscope," *Transducers,* vol. 5, no. 27, pp. 966–968, Nov. 1991.

[11]   N. Yazdi et al., "Micromachined inertial sensors," *Proceedings of the IEEE,* vol. 86, no. 2, pp. 1640–1659, Aug. 1998.

[12]   C. Jiang et al., "Signal processing of MEMS gyroscope arrays to improve accuracy using a 1st order Markov for rate signal modeling," *Sensors*, vol. 12, no. 2, pp. 1720–1737, Mar. 2012.

[13]   J. Mahdi and J. Roshanian, "Optimal redundant sensor configuration for accuracy and reliability increasing in space inertial navigation systems," *The Journal of Navigation,* vol. 66, no. 1, pp. 199–208, Feb. 2013.

[14]   H. D. Escobar Alvarez, "Geometrical configuration comparison of redundant inertial measurement units," M.S. thesis, Dept. Electron. Eng., Univ. of Texas., Austin, TX, 2010.

[15]   J. M. Barrett, "Analyzing and modeling low-cost MEMS IMUs for use in an inertial navigation system," M.S. thesis, Dept. Electron. Eng., Worcester Polytech. Inst., Worcester, MA, 2014.

[16]   The Moore-Penrose pseudoinverse. (n.d.). UCLA. [Online]. Available: http://www.math.ucla.edu/~laub/33a.2.12s/mppseudoinverse.pdf. Accessed Aug. 15, 2014.

[17]   O. Woodman J., "An introduction to inertial navigation," Univ. of Cambridge., Cambridge, UK, Tech. Rep. 696, Aug. 2007.

[18]   W. Stockwell, "Angle random walk," Crossbow Tech. Inc., Milpitas, CA, Tech. Rep. 92–1997, Nov. 2012.

[19]   Descriptive statistics. (n.d.). Univ. of Utah. [Online]. Available: http://www.che.utah.edu/~tony/course/material/Statistics/12_descriptive.php. Accessed Mar. 3, 2014.

[20]   M. Wolf, *Computer as Components Principles of Embedded Computing System Design*, 3rd ed. Waltham, MA: Elsevier Inc., 2012, pp. 1–51.

[21]   Sparkfun 9DOF razor IMU product description page. (n.d.). Sparkfun. [Online]. Available: https://www.sparkfun.com/products/10736. Accessed Jan. 10, 2014.

[22]   National Instruments compact RIO-9024 product description. (n.d.). National Instruments. [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/207371. Accessed Jan. 10, 2014.

[23]   Truncated icosahedron. (n.d.). Wolfram. [Online]. Available: http://mathworld.wolfram.com/TruncatedIcosahedron.html. Accessed Jan. 15, 2014.

[24]   Understanding resolution in high-speed digitizers/oscilloscopes. (Aug. 19, 2014). National Instruments. [Online]. Available: http://www.ni.com/white-paper/4806/en/

[25]   Analog Devices, "Analog Devices digital accelerometer ADXL345," Analog Devices, Norwood, MA, Tech. Rep. ADXL-345, Oct. 2009.

[26] Honeywell, "3-axis digital compass IC HMC5883L advanced information,"
      Honeywell, Sunnyvale, CA, Tech. Rep. HMC-5883L, Mar. 2010.

[27] LabVIEW system design software. (n.d.). National Instruments. [Online].
      Available: http://www.ni.com/labview/. Accessed Mar. 3, 2014.

[28] NX I-deas evolution of excellence. NX Siemens. [Online]. Available:
      http://www.plm.automation.siemens.com/en_us/products/nx/for-simulation/.
      Accessed Feb. 2, 2014.

[29] Future Technology Devices International, "2005FT232R USB UART I.C."
      Future Technology Devices International, Glasgow, UK, Tech. Rep. FT232R,
      Feb. 2005.

[30] RealTerm: serial terminal. (n.d.). RealTerm. [Online]. Available:
      http://RealTerm.sourceforge.net/. Accessed Aug. 3, 2014.

[31] Torsional plant model 205. (n.d.). ECP Systems. [Online]. Available:
      http://www.ecpsystems.com/controls_torplant.htm. Accessed Aug. 3, 2014.

[32] B. Widrow et al., "Statistical theory of quantization," *IEEE Transactions on
      Instrumentation and Measurements,* vol. 45, no. 1, pp. 353–361, Apr. 1996.

[33] Some solid (three-dimensional) geometrical facts about the golden section. (n.d.).
      Univ. of Surrey. [Online]. Available: http://www.maths.surrey.ac.uk/hosted-
      sites/R.Knott/Fibonacci/phi3DGeom.html#icosa. Accessed Sep. 10, 2014.

[34] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd
      ed. San Diego, CA: California Technical Publishing, 1997, pp. 277–285.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1.   Defense Technical Information Center
        Ft. Belvoir, Virginia

2.   Dudley Knox Library
        Naval Postgraduate School
        Monterey, California